



EXCERPTED FROM

STEPHEN  
WOLFRAM  
A NEW  
KIND OF  
SCIENCE

---

NOTES FOR CHAPTER 4:

*Systems Based on  
Numbers*

# Systems Based on Numbers

## The Notion of Numbers

■ **Implementation of digit sequences.** A whole number  $n$  can be converted to a sequence of digits in base  $k$  using `IntegerDigits[n, k]` or (see also page 1094)

```
Reverse[Mod[NestWhileList[Floor[#/k] &, n, # ≥ k &], k]]
```

and from a sequence of digits using `FromDigits[list, k]` or

```
Fold[k #1 + #2 &, 0, list]
```

For a number  $x$  between 0 and 1, the first  $m$  digits in its digit sequence in base  $k$  are given by `RealDigits[x, k, m]` or

```
Floor[k NestList[Mod[k #, 1] &, x, m - 1]]
```

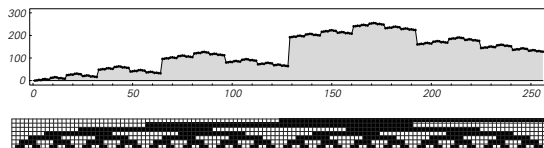
and from these digits one can reconstruct an approximation to the number using `FromDigits[{list, 0}, k]` or

```
Fold[#1/k + #2 &, 0, Reverse[list]]/k
```

■ **Gray code.** In looking at digit sequences, it is sometimes useful to consider ordering numbers by a criterion other than their size. An example is Gray code ordering, in which successive numbers are arranged to differ in only one digit. One possible such ordering for numbers with a total of  $m$  digits is

```
GrayCode[m_] :=  
Nest[Join[#, Length[#] + Reverse[#] &, {0}, m]
```

The succession of sizes and digit sequences of numbers ordered in this way are shown below. (Note that the digit sequence picture is turned on its side relative to those in the main text). The number which appears at position  $i$  is given by `BitXor[i, Floor[i/2]]`. (Iterating the related function `BitXor[i, 2i]` yields numbers whose digit sequences correspond to the rule 60 cellular automaton).



■ **A note for mathematicians.** Some mathematicians will at first find what I say in this chapter quite bizarre. It may help

however to point out that the traditional view of numbers already shows signs of breaking down in many studies of dynamical systems done over the past few decades. Thus for example, instead of getting results in terms of continuous functions, Cantor sets very often appear. Indeed, the symbolic dynamics approach that is often used in dynamical systems theory is quite close to the digit sequence approach I use here—Markov partitions in dynamical systems theory are essentially just generalizations of digit expansions.

However, in the cases that are analyzed in dynamical systems theory, only shifts and other very simple operations are typically performed on digit sequences. And as a result, most of the phenomena that I discuss in this chapter have not been seen in work done in dynamical systems theory.

■ **History of numbers.** Numbers were probably first used many thousands of years ago in commerce, and initially only whole numbers and perhaps rational numbers were needed. But already in Babylonian times, practical problems of geometry began to require square roots. Nevertheless, for a very long time, and despite some development of algebra, only numbers that could somehow in principle be constructed mechanically were ever considered. The invention of fluxions by Isaac Newton in the late 1600s, however, introduced the idea of continuous variables—numbers with a continuous range of possible sizes. But while this was a convenient and powerful notion, it also involved a new level of abstraction, and it brought with it considerable confusion about fundamental issues. In fact, it was really only through the development of rigorous mathematical analysis in the late 1800s that this confusion finally began to clear up. And already by the 1880s Georg Cantor and others had constructed completely discontinuous functions, in which the idea of treating numbers as continuous variables where only the size matters was called into question. But until almost the 1970s, and the emergence of fractal geometry and chaos theory, these functions were largely considered as

mathematical curiosities, of no practical relevance. (See also page 1168.)

Independent of pure mathematics, however, practical applications of numbers have always had to go beyond the abstract idealization of continuous variables. For whether one does calculations by hand, by mechanical calculator or by electronic computer, one always needs an explicit representation for numbers, typically in terms of a sequence of digits of a certain length. (From the 1930s to 1960s, some work was done on so-called analog computers which used electrical voltages to represent continuous variables, but such machines turned out not to be reliable enough for most practical purposes.) From the earliest days of electronic computing, however, great efforts were made to try to approximate a continuum of numbers as closely as possible. And indeed for studying systems with fairly simple behavior, such approximations can typically be made to work. But as we shall see later in this chapter, with more complex behavior, it is almost inevitable that the approximation breaks down, and there is no choice but to look at the explicit representations of numbers. (See also page 1128.)

■ **History of digit sequences.** On an abacus or similar device numbers are in effect represented by digit sequences. In antiquity however most systems for writing numbers were like the Roman one and not based on digit sequences. An exception was the Babylonian base 60 system (from which hours:minutes:seconds notation derives). The Hindu-Arabic base 10 system in its modern form probably originated around 600 AD, and particularly following the work of Leonardo Fibonacci in the early 1200s, became common by the 1400s. Base 2 appears to have first been considered explicitly in the early 1600s (notably by John Napier in 1617), and was studied in detail by Gottfried Leibniz starting in 1679. The possibility of arbitrary bases was stated by Blaise Pascal in 1658. Various bases were used in puzzles, but rarely in pure mathematics (work by Georg Cantor in the 1860s being an exception). The first widespread use of base 2 was in electronic computers, starting in the late 1940s. Even in the 1980s digit sequences were viewed by most mathematicians as largely irrelevant for pure mathematical purposes. The study of fractals and nesting, the appearance of many algorithms involving digit sequences and the routine use of long numbers in *Mathematica* have however gradually made digit sequences be seen as more central to mathematics.

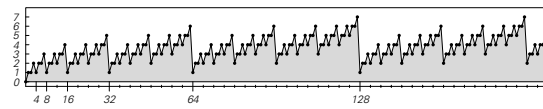
### Elementary Arithmetic

■ **Page 117 · Substitution systems.** There are many connections between digit sequences and substitution systems, as

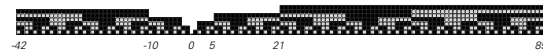
discussed on page 891. The pattern shown here is essentially a rotated version of the pattern generated by the first substitution system on page 83.

■ **Page 117 · Digit counts.** The number of black squares on row  $n$  in the pattern shown here is given by  $DigitCount[n, 2, 1]$  and is plotted below. This function appeared on page 870 in the discussion of binomial coefficients modulo 2, and will appear again in several other places in this book. Note the inequality  $1 \leq DigitCount[n, 2, 1] \leq Log[2, n]$ . Formulas for  $DigitCount[n, 2, 1]$  include  $n - IntegerExponent[n, 2]$  and

$2n - Log[2, Denominator[Derivative[n][1 - \#]^{-1/2} \&][0]/n!]]$   
Straightforward generalizations of  $DigitCount$  can be defined for integer and non-integer bases and by looking not only at the total number of digits but also at correlations between digits. In all cases the analogs of the picture below have a nested structure.

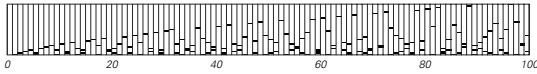


■ **Negative bases.** Given a suitable list of digits from 0 to  $k - 1$  one can obtain any positive or negative number using  $FromDigits[list, -k]$ . The picture below shows the digit sequences of successive numbers in base -2; the row  $j$  from the bottom turns out to consist of alternating black and white blocks of length  $2^j$ . (In ordinary base 2 a number  $-n$  can be represented as on a typical electronic computer by complementing each digit, including leading 0's.) (See also page 1093.)

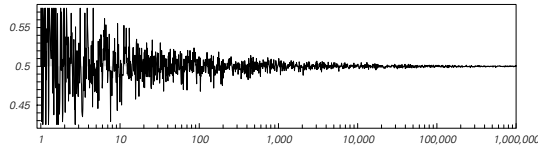


■ **Non-power bases.** One can consider representing numbers by  $Sum[a[n]f[n], \{n, 0, \infty\}]$  where the  $f[n]$  need not be  $k^n$ . So long as  $f[n]$  grows less rapidly than  $2^n$  (as when  $f = Fibonacci$  or  $f = Prime$ ), digits 0 and 1 will suffice, though the representation is not generally unique. (See page 1070.)

■ **Multiplicative digit sequences.** One can consider generalizations of digit sequences in which numbers are broken into parts combined not by addition but by multiplication. Since numbers can be factored uniquely into products of powers of primes, a number can be specified by a list in which 1's appear at the positions of the appropriate  $Prime[m]^n$  (which can be sorted by size) and 0's appear elsewhere, as shown below. Note that unlike the case of ordinary additive digits, far more than  $Log[m]$  digits are required to specify a number  $m$ .



■ **Page 120 · Powers of three in base 2.** The  $n^{\text{th}}$  row in the pattern shown can be obtained simply as `IntegerDigits[3n, 2]`. Even such individual rows seem in many respects random. The picture below shows the fraction of 1's that appear on successive rows. The fraction seems to tend to 1/2.



If one looks only at the rightmost  $s$  columns of the pattern, one sees repetition—but the period of the repetition grows like  $2^s$ . Typical vertical columns have one obvious deviation from randomness: it is twice as probable for the same colors to occur on successive steps than for opposite colors. (For multiplier  $m$  in base  $k$ , the relative frequencies of pairs  $\{i, j\}$  are given by `Quotient[a i - j - 1 + m, k] - Quotient[m i - j - 1, k]`.)

The sequence `Mod[3n, 2s]` obtained from the rightmost  $s$  digits corresponds to a simple linear congruential pseudorandom number generator. Such generators are widely used in practical computer systems, as discussed further on page 974. (Note that in the particular case used here, pairs of numbers `Mod[{3n, 3n+1}, 2s]` always lie on lines; with multipliers other than 3, such regularities may occur for longer blocks of numbers.)

Note that if one uses base 6 rather than base 2, then as shown on page 614 powers of 3 still yield a complicated pattern, but all operations are strictly local, and the system corresponds to a cellular automaton with 6 possible colors for each cell and rule `{a_, b_, c_} → 3 Mod[b, 2] + Floor[c/2]` (see page 1093).

■ **Leading digits.** In base  $b$  the leading digits of powers are not equally probable, but follow the logarithmic law from page 914.

■ **Page 122 · Powers of 3/2.** The  $n^{\text{th}}$  value shown in the plot here is `Mod[(3/2)n, 1]`. Measurements suggest that these values are uniformly distributed in the range 0 to 1, but despite a fair amount of mathematical work since the 1940s, there has been no substantial progress towards proving this.

In base 6, `(3/2)n` is a cellular automaton with rule

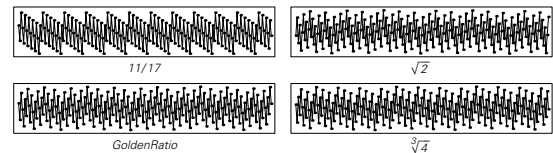
$$\{a_, b_, c_ \} \rightarrow 3 \text{Mod}[a + \text{Quotient}[b, 2], 2] + \text{Quotient}[3 \text{Mod}[b, 2] + \text{Quotient}[c, 2], 2]$$

(Note that this rule is invertible.) Looking at `u(3/2)n` then corresponds to studying the cellular automaton with an initial

condition given by the base 6 digits of  $u$ . It is then possible to find special values of  $u$  (an example is 0.166669170371...) which make the first digit in the fractional part of `u(3/2)n` always nonzero, so that `Mod[u(3/2)n, 1] > 1/6`. In general, it seems that `Mod[u(3/2)n, 1]` can be kept as large as about 0.3 (e.g. with  $u = 0.38906669065 \dots$ ) but no larger.

■ **General powers.** It has been known in principle since the 1930s that `Mod[hn, 1]` is uniformly distributed in the range 0 to 1 for almost all values of  $h$ . However, no specific value of  $h$  for which this is true has ever been explicitly found. (Some attempts to construct such values were made in the 1970s.) Exceptions are known to include so-called Pisot numbers such as `GoldenRatio`,  $\sqrt{2} + 1$  and `Root[#3 - # - 1 &, 1]` (the numerically smallest of all Pisot numbers) for which `Mod[hn, 1]` becomes 0 or 1 for large  $n$ . Note that `Mod[x hn, 1]` effectively extracts successive digits of  $x$  in base  $h$  (see pages 149 and 919).

■ **Multiples of irrational numbers.** Instead of powers one can consider successive multiples `Mod[h n, 1]` of a number  $h$ . The pictures below show results obtained as a function of  $n$  for various choices of  $h$ . (These correspond to positions of a particle bouncing around in an idealized box, as discussed on pages 971 and 1022.)



When  $h$  is a rational number, the sequence always repeats. But in all other cases, the sequence does not repeat, and in fact it is known that a uniform distribution of values is obtained. (The average difference of successive values is maximized for  $h = \text{GoldenRatio}$ , as mentioned on page 891.)

■ **Relation to substitution systems.** Despite the uniform distribution result in the note above, the sequence `Floor[(n + 1)h] - Floor[nh]` is definitely not completely random, and can in fact be generated by a sequence of substitution rules. The first  $m$  rules (which yield far more than  $m$  elements of the original sequence) are obtained for any  $h$  that is not a rational number from the continued fraction form (see page 914) of  $h$  by

$$\text{Map}[\{0 \rightarrow \text{Join}[\#, \{1\}], 1 \rightarrow \text{Join}[\#, \{1, 0\}]\} \&][\text{Table}[0, \{\# - 1\}]] \&, \text{Reverse}[\text{Rest}[\text{ContinuedFraction}[h, m]]]]$$

Given these rules, the original sequence is given by

$$\text{Floor}[h] + \text{Fold}[\text{Flatten}[\#\{1, \#2\} \&, \{0\}], \text{rules}]$$

If  $h$  is the solution to a quadratic equation, then the continued fraction form is repetitive, and so there are a limited number

of different substitution rules. In this case, therefore, the original sequence can be found by a neighbor-independent substitution system of the kind discussed on page 82. For  $h = \text{GoldenRatio}$  the substitution system is  $\{0 \rightarrow \{1\}, 1 \rightarrow \{1, 0\}\}$  (see page 890), for  $h = \sqrt{2}$  it is  $\{0 \rightarrow \{0, 1\}, 1 \rightarrow \{0, 1, 0\}\}$  (see page 892) and for  $h = \sqrt{3}$  it is  $\{0 \rightarrow \{1, 1, 0\}, 1 \rightarrow \{1, 1, 0, 1\}\}$ . (The presence of nested structure is particularly evident in `FoldList[Plus, 0, Table[Mod[h n, 1] - 1/2, {n, max}]]`.) (See also pages 892, 916, 932 and 1084.)

■ **Other uniformly distributed sequences.** Cases in which `Mod[a[n], 1]` is uniformly distributed include  $\sqrt{n}$ ,  $n \text{Log}[n]$ , `Log[Fibonacci[n]]`, `Log[n!]`,  $h n^2$  and  $h \text{Prime}[n]$  ( $h$  irrational) and probably  $n \text{Sin}[n]$ . (See also page 914.)

■ **Page 122 · Implementation.** The evolution for  $t$  steps of the system at the top of the page can be computed simply by

```
NestList[If[EvenQ[#], 3#/2, 3(#+1)/2] &, 1, t]
```

■ **Page 122 · The  $3n+1$  problem.** The system described here is similar to the so-called  $3n+1$  problem, in which one looks at the rule  $n \rightarrow \text{If}[\text{EvenQ}[n], n/2, (3n+1)/2]$  and asks whether for any initial value of  $n$  the system eventually evolves to 1 (and thereafter simply repeats the sequence 1, 2, 1, 2, ...). It has been observed that this happens for all initial values of  $n$  up to at least  $10^{16}$ , but despite a fair amount of mathematical effort since the problem was first posed in the 1930s, no general proof for all values of  $n$  has ever been found. (For negative initial  $n$ , the evolution appears always to reach -1, -5 or -17, and then repeat with periods 1, 3 or 11 respectively.) An alternative formulation is to ask whether for all  $n$

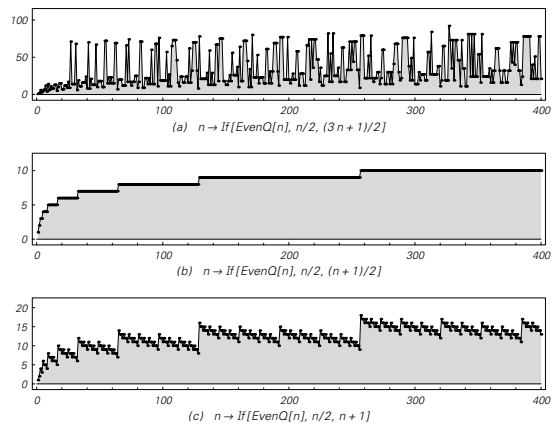
```
FixedPoint[(3#/2^IntegerExponent[# , 2] + 1)/2 &, n] = 2
```

With the rule  $n \rightarrow \text{If}[\text{EvenQ}[n], 5n/2, (n+1)/2]$  used in the main text, the sequence produced repeats if  $n$  ever reaches 2, 4 or 40 (and possibly higher numbers). But with initial values of  $n$  up to 10,000, this happens in only 642 cases, and with values up to 100,000 it happens in only 2683 cases. In all other cases, the values of  $n$  in the sequence appear to grow forever.

To get some idea about the origin of this behavior, one can assume that successive values of  $n$  are randomly even and odd with equal probability. And with this assumption,  $n$  should increase by a factor of 5/2 half the time, and decrease by a factor close to 1/2 the rest of the time—so that after  $t$  steps it should be multiplied by an overall factor of about  $(\sqrt{5}/2)^t$ . Starting with  $n=6$ , the effective exponents for  $t = 10^{\text{Range}[6]}$  are  $\{39.6, 245.1, 1202.8, 9250.7, 98269.8, 1002020.4\}$ . One reason that all sequences do not grow forever is that even with perfect randomness, there will be fluctuations, and occasionally  $n$  will reach a low value that makes it get stuck in a repetitive sequence.

If one applies the same kind of argument to the standard  $3n+1$  problem, then one concludes that  $n$  should on average decrease by a factor of  $\sqrt{3}/2$  at each step, making it unsurprising that at least in most cases  $n$  eventually reaches the value 1. Indeed, averaging over many initial values of  $n$ , there is good quantitative agreement between the predictions of the randomness approximation and the actual  $3n+1$  problem. But since there is no fundamental basis for the randomness approximation, it is still conceivable that a particular value of  $n$  exists that does not follow its predictions.

The pictures below show how many steps are needed to reach value 1 starting from different values of  $n$ . Case (a) is the standard  $3n+1$  problem. Cases (b) and (c) use somewhat different rules that yield considerably simpler behavior. In case (b), the number of steps is equal to the number of base 2 digits in  $n$ , while in case (c) it is determined by the number of 1's in the base 2 digit sequence of  $n$ .



■  **$3n+1$  problem as cellular automaton.** If one writes the digits of  $n$  in base 6, then the rule for updating the digit sequence is a cellular automaton with 7 possible colors (color 6 works as an end marker that appears to the left and right of the actual digit sequence):

```
{a_, b_, c_} -> If[b == 6, If[EvenQ[a], 6, 4],
  3 Mod[a, 2] + Quotient[b, 2] /. 0 -> 6; a == 6]
```

The  $3n+1$  problem can then be viewed as a question about the existence of persistent structure in this cellular automaton.

■ **Reconstructing initial conditions.** Given a particular starting value of  $n$ , it is difficult to predict what precise sequence of even and odd values will be obtained in the system on page 122. But given  $t$  steps in this sequence as a list of 0's and 1's, the

following function will reconstruct the rightmost  $t$  digits in the starting value of  $n$ :

```
IntegerDigits[First[Fold[{Mod[If[OddQ[#2], 2 First[#1] - 1,
  2 First[#1] PowerMod[5, -1, Last[#1]]], Last[#1]],
  2 Last[#1]} &, {0, 2}, Reverse[list]], 2, Length[list]]
```

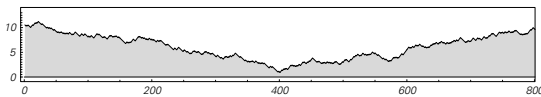
■ **A reversible system.** In both the ordinary  $3n + 1$  problem and in the systems discussed in the main text different numbers often evolve to the same value so that there is no unique way to reverse the evolution. However, with the rule

$$n \rightarrow \text{If}[\text{EvenQ}[n], 3n/2, \text{Round}[3n/4]]$$

it is always possible to go backwards by the rule

$$n \rightarrow \text{If}[\text{Mod}[n, 3] == 0, 2n/3, \text{Round}[4n/3]]$$

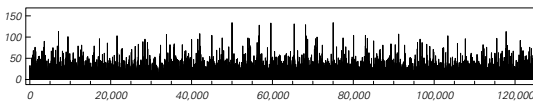
The picture shows the number of base 10 digits in numbers obtained by backward and forward evolution from  $n = 8$ . For  $n < 8$ , the system always enters a short cycle. Starting at  $n = 44$ , there is also a length 12 cycle. But apart from these cycles, the numbers produced always seem to grow without bound at an average rate of  $3/(2\sqrt{2})$  in the forward direction, and  $2 \cdot 4^{1/3}/3$  in the backward direction (at least all numbers up to 10,000 grow to above  $10^{100}$ ). Approximately one number in 20 has the property that evolution either backward or forward from it never leads to a smaller number.



■ **Page 125 · Reversal-addition systems.** The operation that is performed here is

$$n \rightarrow n + \text{FromDigits}[\text{Reverse}[\text{IntegerDigits}[n, 2]], 2]$$

After a few steps, the digit sequence obtained is typically reversal symmetric (a generalized palindrome) except for the interchange of 0 and 1, and for the presence of localized structures. The sequence expands by at least one digit every two steps; more rapid expansion is typically correlated with increased randomness. For most initial  $n$ , the overall pattern obtained quickly becomes repetitive, with an effective period of 4 steps. But with the initial condition  $n = 512$ , no repetition occurs for at least a million steps, at which point  $n$  has 568418 base 2 digits. The plot below shows the lengths of the successive regions of regularity visible on the right-hand edge of the picture on page 126 over the course of the first million steps.



If one works directly with a digit sequence of fixed length, dropping any carries on the left, then a repetitive pattern is typically obtained fairly quickly. If one always includes one

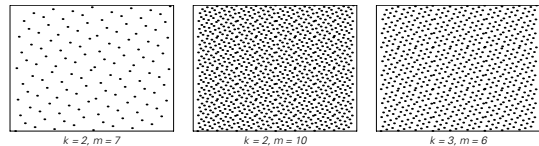
new digit on the left at every step, even when it is 0, then a rather random pattern is produced.

■ **History.** Systems similar to the one described here (though often in base 10) were mentioned in the recreational mathematics literature at least as long ago as 1939. A few small computer experiments were done around 1970, but no large-scale investigations seem to have previously been made.

■ **Digit reversal.** Sequences of the form

```
Table[FromDigits[
  Reverse[IntegerDigits[n, m]], k], {n, 0, k^m - 1}]
```

shown below appear in algorithms such as the fast Fourier transform and, with different values of  $k$  for different coordinates, in certain quasi-Monte Carlo schemes. (See pages 1073 and 1085.) Such sequences were considered by Johannes van der Corput in 1935.



■ **Iterated run-length encoding.** Starting say with  $\{1\}$  consider repeatedly replacing  $list$  by (see page 1070)

```
Flatten[Map[{Length[#], First[#]} &, Split[list]]]
```

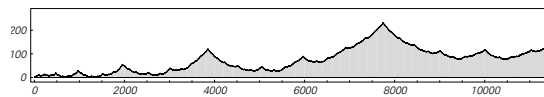
The resulting sequences contain only the numbers 1, 2 and 3, but otherwise at first appear fairly random. However, as noticed by John Conway around 1986, the sequences can actually be obtained by a neighbor-independent substitution system, acting on 92 subsequences, with rules such as  $\{3, 1, 1, 3, 3, 2, 2, 1, 1, 3\} \rightarrow \{\{1, 3, 2\}, \{1, 2, 3, 2, 2, 2, 1, 1, 3\}\}$ .

The system thus in the end produces patterns that are purely nested, though formed from rather complicated elements. The length of the sequence at the  $n^{\text{th}}$  step grows like  $\lambda^n$ , where  $\lambda \approx 1.3$  is the root of a degree 71 polynomial, corresponding to the largest eigenvalue of the transition matrix for the substitution system.

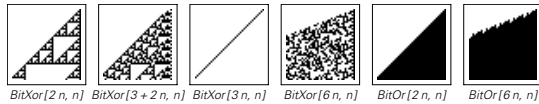
■ **Digit count sequences.** Starting say with  $\{1\}$  repeatedly replace  $list$  by

```
Join[list, IntegerDigits[Apply[Plus, list], 2]]
```

The resulting sequences grow in length roughly like  $n \text{Log}[n]$ . The picture below shows the fluctuations around  $m/2$  of the cumulative number of 1's up to position  $m$  in the sequence obtained at step 1000. A definite nested structure similar to picture (c) on page 130 is evident.



■ **Iterated bitwise operations.** The pictures below show digit sequences generated by repeatedly applying combinations of bitwise and arithmetic operations. The first example corresponds to elementary cellular automaton rule 60. Note that any cellular automaton rule can be reproduced by some appropriate combination of bitwise and arithmetic operations.



### Recursive Sequences

■ **Page 128 · Recurrence relations.** The rules for the sequences given here all have the form of linear recurrence relations. An explicit formula for the  $n^{\text{th}}$  term in each sequence can be found by solving the algebraic equation obtained by applying the replacement  $f[m_] \rightarrow t^m$  to the recurrence relation. (In case (e), for example, the equation is  $t^n = -t^{n-1} + t^{n-2}$ .) Note that (d) is the Fibonacci sequence, discussed on page 890.

Standard examples of recursive sequences that do not come from linear recurrence relations include factorial

$$f[1] = 1; f[n_] := n f[n-1]$$

and Ackermann functions (see below). These two sequences both grow rapidly, but smoothly.

A recurrence relation like

$$f[0] = x; f[n_] := a f[n-1] (1 - f[n-1])$$

corresponds to an iterated map of the kind discussed on page 920, and has complicated behavior for many rational  $x$ .

■ **Ackermann functions.** A convenient example is

$$f[1, n_] := n; f[m_, 1] := f[m-1, 2]$$

$$f[m_, n_] := f[m-1, f[m, n-1] + 1]$$

The original function constructed by Wilhelm Ackermann around 1926 is essentially

$$f[1, x_, y_] := x + y;$$

$$f[m_, x_, y_] := Nest[f[m-1, x, #] &, x, y-1]$$

or

$$f[m_, x_, y_] :=$$

$$Nest[Function[z, Nest[#1, x, z-1]] &, x + # &, m-1][y]$$

For successive  $m$  (following the so-called Grzegorzcyk hierarchy) this is  $x + y$ ,  $xy$ ,  $x^y$ ,  $Nest[x^# &, 1, y]$ , ...  $f[4, x, y]$  can also be written  $\text{Array}[x \&, y, 1, \text{Power}]$  and is sometimes called tetration and denoted  $x \uparrow \uparrow y$ .

■ **Page 129 · Computation of sequences.** It is straightforward to compute the various sequences given here, but to avoid a rapid increase in computer time, it is essential to store all the

values of  $f[n]$  that one has already computed, rather than recomputing them every time they are needed. This is achieved for example by the definitions

$$f[n_] := f[n] = f[n-f[n-1]] + f[n-f[n-2]]$$

$$f[1] = f[2] = 1$$

The question of which recursive definitions yield meaningful sequences can depend on the details of how the rules are applied. For example,  $f[-1]$  may occur, but if the complete expression is  $f[-1] - f[-1]$ , then the actual value of  $f[-1]$  is irrelevant. The default form of evaluation for recursive functions implemented by all standard computer languages (including *Mathematica*) is the so-called leftmost innermost scheme, which attempts to find explicit values for each  $f[k]$  that occurs first, and will therefore never notice if  $f[k]$  in fact occurs only in the combination  $f[k] - f[k]$ . (The SMP system that I built around 1980 allowed different schemes—but they rarely seemed useful and were difficult to understand.)

■ **Page 131 · Properties of sequences.** Sequence (d) is given by

$$f[n_] := (n + g[\text{IntegerDigits}[n, 2]])/2$$

$$g[\{(1)..\}] = 1; g[\{1, (0)..\}] = 0$$

$$g[\{1, s..\}] := 1 + g[\text{IntegerDigits}[\text{FromDigits}\{s\}, 2] + 1, 2]]$$

The list of elements in the sequence up to value  $m$  is given by  $\text{Flatten}[\text{Table}[\text{Table}[n, \{\text{IntegerExponent}[n, 2] + 1\}], \{n, m\}]]$

The differences between the first  $2(2^k - 1)$  of these elements is

$$\text{Nest}[\text{Replace}[\#, \{x\_ \rightarrow \{x, 1, x, 0\}\} \&, \{\}, k]$$

The largest  $n$  for which  $f[n] = m$  is given by  $2m + 1 - \text{DigitCount}[m, 2, 1]$  or  $\text{IntegerExponent}[(2m)!, 2] + 1$  (this satisfies  $h[1] = 2; h[m_] := h[\text{Floor}[m/2] + m]$ ).

The form of sequence (c) is similar to that obtained from concatenation numbers on page 913. Hump  $m$  in the picture of sequence (c) shown is given by

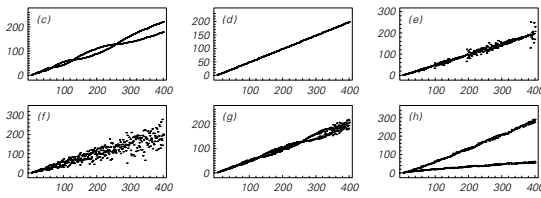
$$\text{FoldList}[\text{Plus}, 0, \text{Flatten}[\text{Nest}[\text{Delete}[\text{NestList}[\text{Rest}, \#, \text{Length}[\#] - 1, 2] \&, \text{Append}[\text{Table}[1, \{m\}], 0], m]] - 1/2]$$

The first  $2^m$  elements in the sequence can also be generated in terms of reordered base 2 digit sequences by

$$\text{FoldList}[\text{Plus}, 1, \text{Map}[\text{Last}[\text{Last}[\#]] \&, \text{Sort}[\text{Table}[\{\{\text{Length}[\#], \text{Apply}[\text{Plus}, \#, 1 - \#]\} \&][\text{IntegerDigits}[i, 2]], \{i, 2^m\}]]]]$$

Note that the positive and negative fluctuations in sequence (f) are not completely random: although the probability for individual fluctuations in each direction seems to be the same, the probability for two positive fluctuations in a row is smaller than for two negative fluctuations in a row.

In the sequences discussed here,  $f[n]$  always has the form  $f[p[n]] + f[q[n]]$ . The plots at the top of the next page show  $p[n]$  and  $q[n]$  as a function of  $n$ .



The process of evaluating  $f[n]$  for a particular  $n$  can be thought of as yielding a tree where each node is a particular  $f[k]$  which has two successors,  $f[p[k]]$  and  $f[q[k]]$ . The distinct nodes reached starting from  $f[12]$  for sequence (f) are then for example  $\{\{12\}, \{3, 7\}, \{1, 2, 4\}, \{1, 2\}, \{1\}\}$ . The total lengths of these chains (corresponding to the depth of the evaluation tree) seem to increase roughly like  $\text{Log}[n]$  for all the rules on this page. For the Fibonacci sequence, it is instead  $n - 1$ . The maximum number of distinct nodes at any level in the tree has large fluctuations but its peaks seem to increase roughly linearly for all the rules on this page (in the Fibonacci case it is  $\text{Ceiling}[n/2]$ ).

■ **History.** The idea of sequences in which later terms are deduced from earlier ones existed in antiquity, notably in the method of induction and in various approximation schemes (compare page 918). The Fibonacci sequence also appears to have arisen in antiquity (see page 890). A fairly clear idea of integer recurrence relations has existed since about the 1600s, but until very recently mainstream mathematics has almost never investigated them. In the late 1800s and early 1900s issues about the foundations of mathematics (see note below) led to the formal definition of so-called recursive functions. But almost without exception the emphasis was on studying what such functions could in principle do, not on looking at the actual behavior of particular ones. And indeed, despite their simple forms, recursive sequences of the kind I discuss here do not for the most part ever appear to have been studied before—although sequence (c) was mentioned in lectures by John Conway around 1988, and the first 17 terms of sequence (e) were given by Douglas Hofstadter in 1979.

■ **Primitive recursive functions.** As part of trying to formalize foundations of arithmetic Richard Dedekind began around 1888 to discuss possible functions that could be defined using recursion (induction). By the 1920s there had then emerged a definite notion of primitive recursive functions. The proof of Gödel's Theorem in 1931 made use of both primitive and general recursive functions—and by the mid-1930s emphasis had shifted to discussion of general recursive functions.

Primitive recursive functions are defined to deal with non-negative integers and to be set up by combining the basic functions  $z = 0$  & (zero),  $s = \# + 1$  & (successor) and

$p[i\_]:= \text{Slot}[i]$  & (projection) using the operations of composition and primitive recursion

```
f[0, y___Integer] := g[y]
f[x_Integer, y___Integer] := h[f[x - 1, y], x - 1, y]
```

Plus and Times can then for example be defined as

```
plus[0, y_] = y; plus[x_, y_] := s[plus[x - 1, y]]
times[0, y_] = 0; times[x_, y_] := plus[times[x - 1, y], y]
```

Most familiar integer mathematical functions also turn out to be primitive recursive—examples being *Power*, *Mod*, *Binomial*, *GCD* and *Prime*. And indeed in the early 1900s it was thought that perhaps any function that could reasonably be computed would be primitive recursive (see page 1125). But the construction in the late 1920s of the Ackermann function  $f[m, x, y]$  discussed above showed that this was not correct. For any primitive recursive function can grow for large  $x$  at most like  $f[m, x, x]$  with fixed  $m$ . Yet  $f[x, x, x]$  will always eventually grow faster than this—demonstrating that the whole Ackermann function cannot be primitive recursive. (See page 1162.)

A crucial feature of primitive recursive functions is that the number of steps they take to evaluate is always limited, and can always in effect be determined in advance, since the basic operation of primitive recursion can be unwound simply as

```
f[x_, y___] := Fold[h[#1, #2, y] &, g[y], Range[0, x - 1]]
```

And what this means is that any computation that for example fundamentally involves a search that might not terminate cannot be implemented by a primitive recursive function. General recursive functions, however, also allow

```
 $\mu[f\_]= \text{NestWhile}[\# + 1 \&, 0, \text{Function}[n, f[n, \#\#1] \neq 0]]$  &
```

which can perform unbounded searches. (Ordinary primitive recursive functions are always total functions, that give definite values for every possible input. But general recursive functions can be partial functions, that do not terminate for some inputs.) As discussed on page 1121 it turns out that general recursive functions are universal, so that they can be used to represent any possible computable function. (Note that any general recursive function can be expressed in the form  $c[f, \mu[g]]$  where  $f$  and  $g$  are primitive recursive.)

In enumerating recursive functions it is convenient to use symbolic definitions for composition and primitive recursion

```
c[g_, h_] = Apply[g, Through[{h}[\#\#]]] &
r[g_, h_] =
  If[\#1 == 0, g[\#\#2], h[\#0[\#1 - 1, \#\#2], \#1 - 1, \#\#2]] &
```

where the more efficient unwound form is

```
r[g_, h_] = Fold[Function[{u, v}, h[u, v, \#\#2]],
  g[\#\#2], Range[0, \# - 1]] &
```

And in terms of these, for example,  $plus = r[p[1], s]$ .



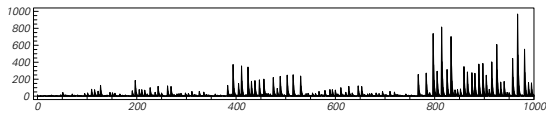
The total number of recursive functions grows roughly exponentially in the size (*LeafCount*) of such expressions, and roughly linearly in the number of arguments.

Most randomly selected primitive recursive functions show very simple behavior—either constant or linearly increasing when fed successive integers as arguments. The smallest examples that show other behavior are:

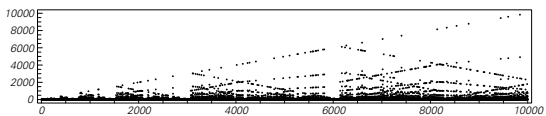
- $r[z, r[s, s]]$ , which is  $1/2\#(\# + 1) \&$ , with quadratic growth
- $r[z, r[s, c[s, s]]]$ , which is  $2^{\#+1} - \# - 2 \&$ , with exponential growth
- $r[z, r[s, p[2]]]$ , which is  $2^{\text{Ceiling}[\text{Log}[2, \# + 2]] - \# - 2 \&$ , which shows very simple nesting
- $r[z, r[c[s, z], z]]$ , which is  $\text{Mod}[\#, 2] \&$ , with repetitive behavior
- $r[z, r[s, r[s, s]]]$  which is  $\text{Fold}[1/2\#1(\#1 + 1) + \#2 \&, 0, \text{Range}[\#]] \&$ , growing like  $2^{2^x}$ .

$r[z, r[s, r[s, r[s, p[2]]]]]$  is the first function to show significantly more complex behavior, and indeed as the picture below indicates, it already shows remarkable randomness.

From its definition, the function can be written as  $\text{Fold}[\text{Fold}[2^{\text{Ceiling}[\text{Log}[2, \text{Ceiling}[(\#1 + 2)/(\#2 + 2)]]}(\#2 + 2) - 2 - \#1 \&, \#2, \text{Range}[\#1]] \&, 0, \text{Range}[\#]] \&$ . Its first zeros are at  $\{4, 126, 813, 966, 1166, 1177, 1666, 1897\}$ .



Each zero is immediately followed by a maximum equal to  $x$ , and as picture below shows, values tend to accumulate for example on lines of the form  $\pm x/2^u \pm (2m + 1)2^v$ .

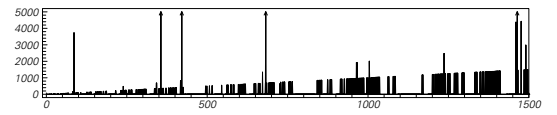


Note that functions of the form  $\text{Nest}[r[c[s, z], \#] \&, c[s, s], n]$  are given in terms of the original Ackermann function in the note above by  $f[n + 1, 2, \# + 1] - 1 \&$ .

Before the example above one might have thought that primitive recursive functions would always have to show rather simple behavior. But already an immediate counterexample is *Prime*. And it turns out that if they never sample values below  $f[0]$  the functions in the main text are also all primitive recursive. (Their definitions have a

primitive recursive structure, but to operate correctly they must be given integers that are non-negative.)

Among functions with simple explicit definitions, essentially the only examples known fundamentally to be not primitive recursive are ones closely related to the Ackermann function. But given an enumeration of primitive recursive functions (say ordered first by *LeafCount*, then with *Sort*) in which the  $m^{\text{th}}$  function is  $w[m]$  diagonalization (see page 1128) yields the function  $w[x][x]$  shown below which cannot be primitive recursive. It is inevitable that the function shown must eventually grow faster than any primitive recursive function (at  $x = 356$  its value is 63190, while at  $x = 1464$  it is 1073844). But by reducing the results modulo 2 one gets a function that does not grow—and has seemingly quite random behavior—yet is presumably again not primitive recursive.

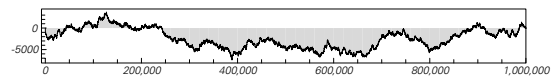


(Note that multiple arguments to a recursive function can be encoded as a single argument using functions like the  $\beta$  of page 1120—though the irregularity of such functions tends to make it difficult then to tell what is going on in the underlying recursive function.)

▪ **Ulam sequences.** Slightly more complicated definitions in terms of numbers yield all sorts of sequences with very complicated forms. An example suggested by Stanislaw Ulam around 1960 (in a peculiar attempt to get a 1D analog of a 2D cellular automaton; see pages 877 and 928) starts with  $\{1, 2\}$ , then successively appends the smallest number that is the sum of two previous numbers in just one way, yielding

- $\{1, 2, 3, 4, 6, 8, 11, 13, 16, 18, 26, 28, 36, 38, 47, 48, 53, 57, \dots\}$

With this initial condition, the sequence is known to go on forever. At least up to  $n = 10^6$  terms, it increases roughly like  $13.5n$ , but as shown below the fluctuations seem random.



**The Sequence of Primes**

▪ **History of primes.** Whether the Babylonians had the notion of primes is not clear, but before 400 BC the Pythagoreans had introduced primes as numbers of objects that can be

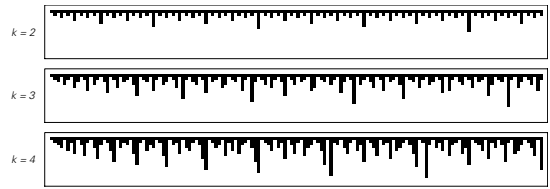
arranged only in a single line, and not in any other rectangular array. Around 300 BC Euclid discussed various properties of primes in his *Elements*, giving for example a proof that there are an infinity of primes. The sieve of Eratosthenes was described in 200 BC, apparently following ideas of Plato. Then starting in the early 1600s various methods for factoring were developed, and conjectures about formulas for primes were made. Pierre Fermat suggested  $2^{2^n} + 1$  as a source for primes and Marin Mersenne  $2^{\text{Prime}[n]} - 1$  (see page 911). In 1752 Christian Goldbach showed that no ordinary polynomial could generate only primes, though as pointed out by Leonhard Euler  $n^2 - n + 41$  does so for  $n < 40$ . (With *If* or *Floor* included there are at least complicated cases known where polynomial-like formulas can be set up whose evaluation corresponds to explicit prime-generating procedures—see page 1162.) Starting around 1800 extensive work was done on analytical approximations to the distribution of primes (see below). There continued to be slow progress in finding specific large primes;  $2^{31} - 1$  was found prime around 1750 and  $2^{127} - 1$  in 1876. ( $2^{25} + 1$  was found composite in 1732, as have now all  $2^{2^n} + 1$  for  $n \leq 32$ .) Then starting in the 1950s with the use of electronic computers many new large primes were found. The number of digits in the largest known prime has historically increased roughly exponentially with time over the past two decades, with a prime of over 4 million digits ( $2^{13466917} - 1$ ) now being known (see page 911).

■ **Page 132 · Finding primes.** The sieve of Eratosthenes shown in the picture is an appropriate procedure if one wants to find every prime, but testing whether an individual number is prime can be done much more efficiently, as in *PrimeQ[n]* in *Mathematica*, for example by using Fermat's so-called little theorem that  $\text{Mod}[a^{p-1}, p] = 1$  whenever  $p$  is prime. The  $n^{\text{th}}$  prime *Prime[n]* can also be computed fairly efficiently using ideas from analytic number theory (see below).

■ **Decimation systems.** A somewhat similar system starts with a line of cells, then at each step removes every  $k^{\text{th}}$  cell that remains, as in the pictures below. The number of steps for which a cell at position  $n$  will survive can be computed as

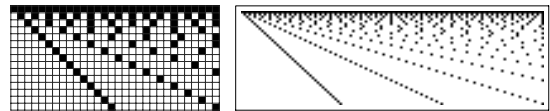
$$\text{Module}[\{q = n + k - 1, s = 1\}, \\ \text{While}[\text{Mod}[q, k] \neq 0, q = \text{Ceiling}[(k - 1)q/k]; s++]; s]$$

If a cell is going to survive for  $s$  steps, then it turns out that this can be determined by looking at the last  $s$  digits in the base  $k$  representation of its position. For  $k = 2$ , a cell survives for  $s$  steps if these digits are all 0 (so that  $s = \text{IntegerExponent}[n, k]$ ). But for  $k > 2$ , no such simple characterization appears to exist.



If the cells are arranged on a circle of size  $n$ , the question of which cell is removed last is the so-called Josephus problem. The solution is  $\text{Fold}[\text{Mod}[\#1 + k, \#2, 1] \&, 0, \text{Range}[n]]$ , or  $\text{FromDigits}[\text{RotateLeft}[\text{IntegerDigits}[n, 2]]]$  for  $k = 2$ .

■ **Page 132 · Divisors.** The picture below shows as black squares the divisors of each successive number (which correspond to the gray dots in the picture in the main text). Primes have divisors 1 and  $n$  only. (See also pages 902 and 747.)



■ **Page 133 · Results about primes.** *Prime[n]* is given approximately by  $n \text{Log}[n] + n \text{Log}[\text{Log}[n]]$ . (*Prime[10^9]* is 22,801,763,489 while the approximation gives  $2.38 \times 10^{10}$ .) A first approximation to *PrimePi[n]* is  $n/\text{Log}[n]$ . A somewhat better approximation is *LogIntegral[n]*, equal to  $\text{Integrate}[1/\text{Log}[t], \{t, 2, n\}]$ . This was found empirically by Carl Friedrich Gauss in 1792, based on looking at a table of primes. (*PrimePi[10^9]* is 50,847,534 while *LogIntegral[10^9]* is about 50,849,235.) A still better approximation is obtained by subtracting  $\text{Sum}[\text{LogIntegral}[n^{r_i}], \{i, -\infty, \infty\}]$  where the  $r_i$  are the complex zeros of the Riemann zeta function *Zeta[s]*, discussed on page 918. According to the Riemann Hypothesis, the difference between *PrimePi[n]* and *LogIntegral[n]* is of order  $\sqrt{n} \text{Log}[n]$ . More refined analytical estimates of *PrimePi[n]* are good enough that they are used by *Mathematica* to compute *Prime[n]* for large  $n$ .

It is known that the ratio of the number of primes of the form  $4k + 1$  and  $4k + 3$  asymptotically approaches 1, but almost nothing has been proved about the fluctuations.

The gap between successive primes  $\text{Prime}[n] - \text{Prime}[n - 1]$  is thought to grow on average at most like  $\text{Log}[\text{Prime}[n]]^2$ . It is known that for sufficiently large  $n$  a gap of any size must exist. It is believed but not proved that there are an infinite number of "twin primes" with a gap of exactly 2.

■ **History of number theory.** Most areas of mathematics go from inception to maturity within at most a century. But in

number theory there are questions that were formulated more than 2000 years ago (such as whether any odd perfect numbers exist) that have still not been answered. Of the principles that have been established in number theory, a great many were first revealed by explicit experiments. From its inception in classical times, through its development in the 1600s to 1800s, number theory was largely separate from other fields of mathematics. But starting at the end of the 1800s, increasing connections were found to other areas of both continuous and discrete mathematics. And through these connections, sophisticated proofs of such results as Fermat's Last Theorem—open for 350 years—have been constructed. Long considered a rather esoteric branch of mathematics, number theory has in recent years grown in practical importance through its use in areas such as coding theory, cryptography and statistical mechanics. Properties of numbers and certain elementary aspects of number theory have also always played a central role in amateur and recreational mathematics. And as this chapter indicates, number theory can also be used to provide many examples of the basic phenomena discussed in this book.

■ **Page 134 · Tables of primes.** No explicit tables of primes appear to have survived from antiquity, but it seems likely that all primes up to somewhere between 5000 and 10000 were known. (In 348 BC, Plato mentioned divisors of 5040, and by 100 AD there is evidence that the fifth perfect number was known, requiring the knowledge that 8191 is prime.) In 1202 Leonardo Fibonacci explicitly gave an example a list of primes up to 100. And by the mid-1600s there were printed tables of primes up to 100,000, containing as much data as in plots (c) and (d). In the 1700s and 1800s many tables of number factorizations were constructed; by the 1770s there was a table up to 2 million, and by the 1860s up to 100 million. A table of primes up to a trillion could now be generated fairly easily with current computer technology—though for most purposes computation of specific primes is more useful.

■ **Page 134 · Numbers of primes.** The fact that curve (c) must cross the axis was proved by John Littlewood in 1914, and it is known to have at least one crossing below  $10^{317}$ . Somewhat related to the curves shown here is the function  $MoebiusMu[n]$ , equal to 0 if  $n$  has a repeated prime factor and otherwise  $(-1)^{Length[FactorInteger[n]]}$ . The quantity  $FoldList[Plus, 0, Table[MoebiusMu[i], {i, n}]]$  behaves very much like a random walk. The so-called Mertens Conjecture from 1897 stated that the magnitude of this quantity is less than  $\sqrt{n}$ . But this was disproved in 1983, although the necessary  $n$  is not known explicitly.

■ **Relative primes.** A single number is prime if it has no non-trivial factors. Two numbers are said to be relatively prime if they share no non-trivial factors. The pattern formed by numbers with this property is shown on page 613.

■ **Page 135 · Properties.** (a) The number of divisors of  $n$  is given by  $DivisorSigma[0, n]$ , equal to  $Length[Divisors[n]]$ . For large  $n$  this number is on average of order  $Log[n] + 2 EulerGamma - 1$ .

(b) (*Aliquot sums*) The quantity that is plotted is  $DivisorSigma[1, n] - 2n$ , equal to  $Apply[Plus, Divisors[n]] - 2n$ . This quantity was considered of great significance in antiquity, particularly by the Pythagoreans. Numbers were known as abundant, deficient or perfect depending on whether the quantity was positive, negative or zero. (See notes on perfect numbers below.) For large  $n$ ,  $DivisorSigma[1, n]$  is known to grow at most like  $Log[Log[n]] n Exp[EulerGamma]$ , and on average like  $\pi^2 n/6$  (see page 1093). As discovered by Srinivasa Ramanujan in 1918 its fluctuations (see below) can be obtained from the formula

$$\frac{1}{6} \pi^2 n \text{Sum}[Apply[Plus, Cos[2 \pi n \text{Select}[Range[s], GCD[s, \#] == 1 \&]/s]]/s^2, \{s, \infty\}]$$

(c) Squares are taken to be of positive or negative integers, or zero. The number of ways of expressing an integer  $n$  as the sum of two such squares is  $4 \text{Apply}[Plus, Im[i^{\wedge} Divisors[n]]]$ . This is nonzero when all prime factors of  $n$  of the form  $4k + 3$  appear with even exponents. There is no known simple formula for the number of ways of expressing an integer as a sum of three squares, although part of the condition in the main text for integers to be expressible in this way was established by René Descartes in 1638 and the rest by Adrien Legendre in 1798. Note that the total number of integers less than  $n$  which can be expressed as a sum of three squares increases roughly like  $5n/6$ , with fluctuations related to  $IntegerDigits[n, 4]$ . It is known that the directions of all vectors  $\{x, y, z\}$  for which  $x^2 + y^2 + z^2 = n$  are uniformly distributed in the limit of large  $n$ .

The total number of ways that integers less than  $n$  can be expressed as a sum of  $d$  squares is equal to the number of integer lattice points that lie inside a sphere of radius  $\sqrt{n}$  in  $d$ -dimensional space. For  $d = 2$ , this approaches  $\pi n$  for large  $n$ , with an error of order  $n^c$ , where  $1/4 < c \leq 0.315$ .

(d) All numbers  $n$  can be expressed as the sum of four squares, in exactly  $8 \text{Apply}[Plus, Select[Divisors[n], Mod[\#, 4] \neq 0 \&]]$  ways, as established by Carl Jacobi in 1829. Edward Waring stated in 1770 that any number can be expressed as a sum of at most 9 cubes and 19 fourth powers. Seven cubes appear to suffice for all but 17 numbers, the last of which is 455; four

cubes may suffice for all but 113936676 numbers, the last of which is 7373170279850. (See also page 1166.)

(e) Goldbach's Conjecture has been verified for all even numbers up to  $4 \times 10^{14}$ . In 1973 it was proved that any even number can be written as the sum of a prime and a number that has at most two prime factors, not necessarily distinct. The number of ways of writing an integer  $n$  as a sum of two primes can be calculated explicitly as `Length[Select[n - Table[Prime[i], {i, PrimePi[n]}], PrimeQ]]`.

This quantity was conjectured by G. H. Hardy and John Littlewood in 1922 to be proportional to

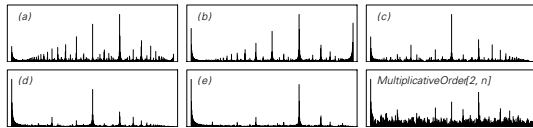
$$2n \text{Apply}[Times, Map[(\# - 1)/(\# - 2) \&, Map[First, Rest[FactorInteger[n]]]]]/Log[n]^2$$

It was proved in 1937 by Ivan Vinogradov that any large odd integer can be expressed as a sum of three primes.

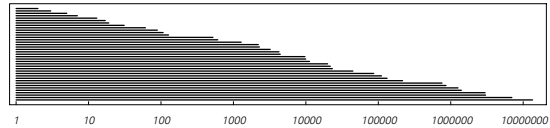
■ **Trapezoidal primes.** If one lays out  $n$  objects in an  $a \times b$  rectangular array, then  $n$  is prime if either  $a$  or  $b$  must be 1. Following the Pythagorean idea of figurate numbers one can instead consider laying out objects in an array of  $b$  rows, containing successively  $a, a - 1, \dots$  objects. It turns out all numbers except powers of 2 can be represented this way.

■ **Other integer functions.** `IntegerExponent[n, k]` gives nested behavior as for decimation systems on page 909, while `MultiplicativeOrder[k, n]` and `EulerPhi[n]` yield more complicated behavior, as shown on pages 257 and 1093.

■ **Spectra.** The pictures below show frequency spectra obtained from the sequences in the main text. Some regularity is evident, and in cases (a) and (b) it can be understood from trigonometric sum formulas of Ramanujan discussed above (see also pages 586 and 1081).

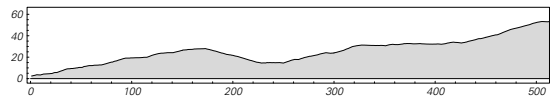


■ **Perfect numbers.** Perfect numbers with the property that `Apply[Plus, Divisors[n]] == 2n` have been studied since at least the time of Pythagoras around 500 BC. The first few perfect numbers are  $\{6, 28, 496, 8128, 33550336\}$  (a total of 39 are currently known). It was shown by Euclid in 300 BC that  $2^{n-1}(2^n - 1)$  is a perfect number whenever  $2^n - 1$  is prime. Leonhard Euler then proved around 1780 that every even perfect number must have this form. The values of  $n$  for the known Mersenne primes  $2^n - 1$  are shown below. These values can be found using the so-called Lucas-Lehmer test `Nest[Mod[#^2 - 2, 2^n - 1] \&, 4, n - 2] == 0`, and in all cases  $n$  itself must be prime.



Whether any odd perfect numbers exist is probably the single oldest unsolved problem in mathematics. It is known that any odd perfect number must be greater than  $10^{300}$ , must have a factor of at least  $10^6$ , and must be less than  $4^{4^s}$  if it has only  $s$  prime factors. Looking at curve (b) on page 135, however, it does not seem inconceivable that an odd perfect number could exist. For odd  $n$  up to 500 million the only values near 0 that appear in the curve are  $\{-6, -5, -4, -2, -1, 6, 18, 26, 30, 36\}$ , with, for example, the first 6 occurring at  $n = 8925$  and last 18 occurring at  $n = 159030135$ . Various generalizations of perfect numbers have been considered, requiring for example `IntegerQ[DivisorSigma[1, n]/n]` (pluperfect) or `Abs[DivisorSigma[1, n] - 2n] < r` (quasiperfect).

■ **Iterated aliquot sums.** Related to case (b) above is a system which repeats the replacement  $n \rightarrow \text{Apply}[Plus, Divisors[n]] - n$  or equivalently  $n \rightarrow \text{DivisorSigma}[1, n] - n$ . The fixed points of this procedure are the perfect numbers (see above). Other numbers usually evolve to perfect numbers, or to short repetitive sequences of numbers. But if one starts, for example, with the number 276, then the picture below shows the number of base 10 digits in the value obtained at each step.



After 500 steps, the value is the 53-digit number  
 $39448887705043893375102470161238803295318090278129552$

The question of whether such values can increase forever was considered by Eugène Catalan in 1887, and has remained unresolved since.

**Mathematical Constants**

■ **Page 137 · Digits of pi.** The digits of  $\pi$  shown here can be obtained in less than a second from `Mathematica` on a typical current computer using `N[\pi, 7000]`. Historically, the number of decimal digits of  $\pi$  that have been computed is roughly as follows: 2000 BC (Babylonians, Egyptians): 2 digits; 200 BC (Archimedes): 5 digits; 1430 AD: 14 digits; 1610: 35 digits; 1706: 100 digits; 1844: 200 digits; 1855: 500 digits; 1949 (ENIAC computer): 2037 digits; 1961: 100,000 digits (IBM 7090); 1973: 1 million; 1983: 16 million; 1989: 1 billion; 1997: 50 billion; 1999: 206 billion. In the first 200

billion digits, the frequencies of 0 through 9 differ from 20 billion by

```
{30841, -85289, 136978, 69393, -78309,
-82947, -118485, -32406, 291044, -130820}
```

An early approximation to  $\pi$  was

```
4 Sum[(-1)^k/(2k + 1), {k, 0, m}]
```

30 digits were obtained with

```
2 Apply[Times, 2/Rest[NestList[Sqrt[2 + #] &, 0, m]]]
```

An efficient way to compute  $\pi$  to  $n$  digits of precision is

```
(#[[2]]^2/#[[3]] &)[NestWhile[Apply[Function[{a, b, c, d},
{(a + b)/2, Sqrt[a b], c - d (a - b)^2, 2 d}], #] &,
{1, 1/Sqrt[N[2, n]], 1/4, 1/4}, #[[1]] ##[[2]] &]]]
```

This requires about  $\text{Log}[2, n]$  steps, or a total of roughly  $n \text{Log}[n]^2$  operations (see page 1134).

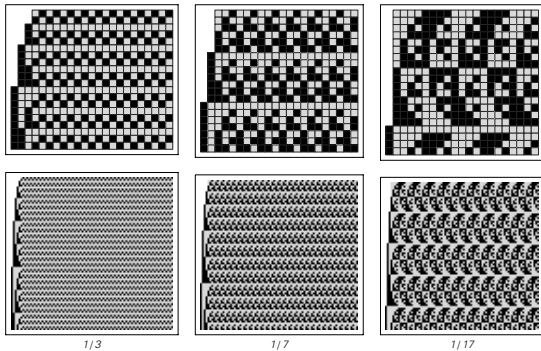
■ **Computing  $n^{\text{th}}$  digits directly.** Most methods for computing mathematical constants progressively generate each additional digit. But following work by Simon Plouffe and others in 1995 it became clear that it is sometimes possible to generate, at least with overwhelming probability, the  $n^{\text{th}}$  digit without explicitly finding previous ones. As an example, the  $n^{\text{th}}$  digit of  $\text{Log}[2]$  in base 2 is formally given by  $\text{Round}[\text{FractionalPart}[2^n \text{Sum}[2^{-k}/k, \{k, \infty\}]]]$ . And in practice the  $n^{\text{th}}$  digit can be found just by computing slightly over  $n$  terms of the sum, according to

```
Round[FractionalPart[
Sum[FractionalPart[PowerMod[2, n - k, k]/k], {k, n}] +
Sum[2^{n-k}/k, {k, n + 1, n + d}]]]
```

where several values of  $d$  can be tried to check that the result does not change. (Note that with finite-precision arithmetic, some exponentially small probability exists that truncation of numbers will lead to incorrect results.) The same basic approach as for  $\text{Log}[2]$  can be used to obtain base 16 digits in  $\pi$  from the following formula for  $\pi$ :

```
Sum[16^{-k} (4/(8k + 1) - 2/(8k + 4) -
1/(8k + 5) - 1/(8k + 6)), {k, 0, \infty}]
```

A similar approach can also be used for many other constants that can be viewed as related to values of  $\text{PolyLog}$ .



■ **Page 139 • Rational numbers.** The pictures above show the base 2 digit sequences of numbers  $m/n$  for successive  $m$ .

The digits of  $1/n$  in base  $b$  repeat with period

```
MultiplicativeOrder[b, FixedPoint[#/GCD[#, b] &, n]]
```

which is equal to  $\text{MultiplicativeOrder}[b, n]$  for prime  $n$ , and is at most  $n - 1$ . Each repeating block of digits typically seems quite random, and has properties such as all possible subblocks of digits up to a certain length appearing (see page 1084).

■ **Page 139 • Digit sequence properties.** Empirical evidence for the randomness of the digit sequences of  $\sqrt{n}$ ,  $\pi$ , etc. has been accumulating since early computer experiments in the 1940s. The evidence is based on applying various standard statistical tests of randomness, and remains somewhat haphazard. (Already in 1888 John Venn had noted for example that the first 707 digits of  $\pi$  lead to an apparently typical 2D random walk.) (See page 1089.)

The fact that  $\sqrt{2}$  is not a rational number was discovered by the Pythagoreans. Numbers that arise as solutions of polynomial equations are called algebraic; those that do not are called transcendental.  $e$  and  $\pi$  were proved to be transcendental in 1873 and 1882 respectively. It is known that  $\text{Exp}[n]$  and  $\text{Log}[n]$  for whole numbers  $n$  (except 0 and 1 respectively) are transcendental. It is also known for example that  $\text{Gamma}[1/3]$  and  $\text{BesselJ}[0, n]$  are transcendental. It is not known for example whether  $\text{EulerGamma}$  is even irrational.

A number is said to be “normal” in a particular base if every digit and every block of digits of any length occur with equal frequency. Note that the fact that a number is normal in one base does not imply anything about its normality in another base (unless the bases are related for example by both being powers of 2). Despite empirical evidence, no number expressed just in terms of standard mathematical functions has ever been rigorously proved to be normal. It has nevertheless been known since the work of Emile Borel in 1909 that numbers picked randomly on the basis of their value are almost always normal. And indeed with explicit constructions in terms of digits, it is quite straightforward to get numbers that are normal. An example of this is the number 0.1234567891011121314... obtained by concatenating the digits of successive integers in base 10 (see below). This number was discussed by David Champernowne in 1933, and is known to be transcendental. A few other results are also known. One based on gradual extension of work by Richard Stoneham from 1971 is that numbers of the form  $\text{Sum}[1/(p^n b^{p^n}), \{n, \infty\}]$  for prime  $p > 2$  are normal in base  $b$  (for  $\text{GCD}[b, p] = 1$ ), and are transcendental.

■ **Page 141 · Square roots.** A standard way to compute  $\sqrt{n}$  is Newton's method (actually used already in 2000 BC by the Babylonians), in which one takes an estimate of the value  $x$  and then successively applies the rule  $x \rightarrow 1/2(x + n/x)$ . After  $t$  steps, this method yields a result accurate to about  $t^2$  digits.

Another approach to computing square roots is based on the fact that the ratio of successive terms in for example the sequence  $f[i] = 2f[i-1] + f[i-2]$  with  $f[1] = f[2] = 1$  tends to  $1 + \sqrt{2}$ . This method yields about 2.5  $t$  base 2 digits after  $t$  steps.

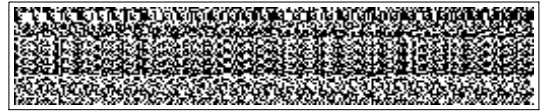
The method of computing square roots shown in the main text is less efficient (it computes  $t$  digits in  $t$  steps), but illustrates more of the mechanisms involved. The basic idea is at every step  $t$  to maintain the relation  $s^2 + 4r = 4^t n$ , keeping  $r$  as small as possible so as to make  $s \leq 2^t \sqrt{n} < s + 4$ . Note that the method works not only for integers, but for any rational number  $n$  for which  $1 \leq n < 4$ .

■ **Nested digit sequences.** The number obtained from the substitution system  $\{1 \rightarrow \{1, 0\}, 0 \rightarrow \{0, 1\}\}$  is approximately 0.587545966 in base 10. It is certainly conceivable that a quantity such as Feigenbaum's constant (approximately 4.6692016091) could have a digit sequence with this kind of nested structure.

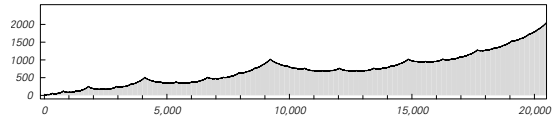
From the result on page 890, the number whose digits are obtained from  $\{1 \rightarrow \{1, 0\}, 0 \rightarrow \{1\}\}$  is given by  $\text{Sum}[2^{-(\text{Floor}[n \text{ GoldenRatio}])}, \{n, \infty\}]$ . This number is known to be transcendental. The  $n^{\text{th}}$  term in its continued fraction representation turns out to be  $2^{\wedge} \text{Fibonacci}[n-2]$ .

The fact that nested digit sequences do not correspond to algebraic numbers follows from work by Alfred van der Poorten and others in the early 1980s. The argument is based on showing that an algebraic function always exists for which the coefficients in its power series correspond to any given nested sequence when reduced modulo some  $p$ . (See page 1092.) But then there is a general result that if a particular sequence of power series coefficients can be obtained from an algebraic (but not rational) function modulo a particular  $p$ , then it can only be obtained from transcendental functions modulo any other  $p$ —or over the integers.

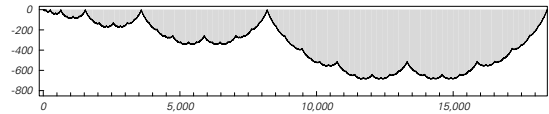
■ **Concatenation sequences.** One can consider forming sequences by concatenating digits of successive integers in base  $k$ , as in  $\text{Flatten}[\text{Table}[\text{IntegerDigits}[i, k], \{i, n\}]]$ . In the limit, such sequences contain with equal frequency all possible blocks of any given length, but as shown on page 597, they exhibit other obvious deviations from randomness. The picture below shows the  $k = 2$  sequence chopped into length 256 blocks.



Applying  $\text{FoldList}[\text{Plus}, 0, 2 \text{ list} - 1]$  to the whole sequence yields the pattern shown below.



The systematic increase is a consequence of the leading 1 in each concatenated sequence. Dropping this 1 yields the pattern below.



This is similar to picture (c) on page 131, and is a digit-by-digit version of

```
FoldList[Plus, 0,
Table[Apply[Plus, 2 Rest[IntegerDigits[i, 2]] - 1], {i, n}]]
```

Note that although the picture above has a nested structure, the original concatenation sequences are not nested, and so cannot be generated by substitution systems. The element at position  $n$  in the first sequence discussed above can however be obtained in about  $\text{Log}[n]$  steps using

```
((IntegerDigits[#3 + Quotient[#1, #2], 2])[[
Mod[#1, #2] + 1] &])[n - (# - 2) 2^{#-1} - 2, #,
2^{#-1}] &][NestWhile[# + 1 &, 0, (# - 1) 2^{#} + 1 < n &]]
```

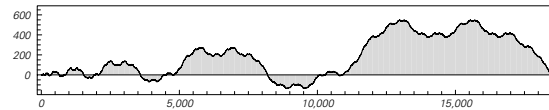
where the result of the  $\text{NestWhile}$  can be expressed as

```
Ceiling[1 + ProductLog[1/2 (n - 1) Log[2]]/Log[2]]
```

Following work by Maxim Rytin in the late 1990s about  $k^{n+1}$  digits of a concatenation sequence can be found fairly efficiently from

$$\frac{k}{(k-1)^2} - \frac{(k-1) \text{Sum}[k^{(k^s-1)(1+s-k)/(k-1)} \{1/((k-1)(k^s-1)^2) - k/((k-1)(k^{s+1}-1)^2) + 1/(k^{s+1}-1)\}, \{s, n\}]}{k^{(k^s-1)(1+s-k)/(k-1)}}$$

Concatenation sequences can also be generated by joining together digits from other representations of numbers; the picture below shows results for the Gray code representation from page 901.



■ **Specially constructed transcendental numbers.** Numbers known to be transcendental include ones whose digit sequences contain 1's only at positions  $n!$ ,  $2^n$  or *Fibonacci*[ $n$ ]. Concatenation sequences, as well as generalizations formed by concatenating values of polynomials at successive integer points, are also known to yield numbers that are transcendental.

■ **Runs of digits.** One can consider any base 2 digit sequence as consisting of successive runs of 0's and 1's, constructed from the list of run lengths by

```
Fold[Join[#1, Table[1 - Last[#1], {#2}]] &, {0}, list]
```

This representation is related to so-called surreal numbers (though with the first few digits different). The number with run lengths corresponding to successive integers (so that the  $n^{\text{th}}$  digit is  $\text{Mod}[\text{Floor}[1/2 + \text{Sqrt}[2n]], 2]$ ) turns out to be  $(1 - 2^{1/4} \text{EllipticTheta}[2, 0, 1/2] + \text{EllipticTheta}[3, 0, 1/2])/2$ , and appears at least not to be algebraic.

■ **Leading digits.** Even though in individual numbers generated by simple mathematical procedures all possible digits often appear to occur with equal frequency, leading digits in sequences of numbers typically do not. Instead it is common for a leading digit  $s$  in base  $b$  to occur with frequency  $\text{Log}[b, (s + 1)/s]$  (so that in base 10 1's occur 30% of the time and 9's 4.5%). This will happen whenever  $\text{FractionalPart}[\text{Log}[b, a[n]]]$  is uniformly distributed, which, as discussed on page 903, is known to be true for sequences such as  $r^n$  (with  $\text{Log}[b, r]$  irrational),  $n^n$ ,  $n!$ , *Fibonacci*[ $n$ ], but not  $r n$ , *Prime*[ $n$ ] or  $\text{Log}[n]$ . A logarithmic law for leading digits is also found in many practical numerical tables, as noted by Simon Newcomb in 1881 and Frank Benford in 1938.

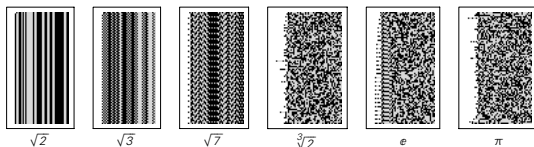
■ **Page 143 • Continued fractions.** The first  $n$  terms in the continued fraction representation for a number  $x$  can be found from the built-in *Mathematica* function *ContinuedFraction*, or from

```
Floor[NestList[1/Mod[#1, 1] &, x, n - 1]]
```

A rational approximation to the number  $x$  can be reconstructed from the continued fraction using *FromContinuedFraction* or by

```
Fold[1/#1 + #2 &, Last[list], Rest[Reverse[list]]]
```

The pictures below show the digit sequences of successive iterates obtained from  $\text{NestList}[1/\text{Mod}[\#, 1] \&, x, n]$  for several numbers  $x$ .



Unlike ordinary digits, the individual terms in a continued fraction can be of any size. In the continued fraction for a randomly chosen number, the probability to find a term of size  $s$  is  $\text{Log}[2, (1 + 1/s)/(1 + 1/(s + 1))]$ , so that the probability of getting a 1 is about 41.50%, and the probability of getting a large term falls off like  $1/s^2$ . If one looks at many terms, then their geometric mean is finite, and approaches Khinchin's constant *Khinchin*  $\approx 2.68545$ .

In the first 1000 terms of the continued fraction for  $\pi$ , there are 412 1's, and the geometric mean is about 2.6656. The largest individual term is the 432th one, which is equal to 20,776. In the first million terms, there are 414,526 1's, the geometric mean is 2.68447, and the largest term is the 453,294th one, which is 12,996,958.

Note that although the usual continued fraction for  $\pi$  looks quite random, modified forms such as

```
4/(Fold[#2/#1 + 2 &, 2, Reverse[Range[1, n, 2]^2]] - 1)
```

can be very regular.

The continued fractions for  $\text{Exp}[2/k]$  and  $\text{Tan}[k/2]$  have simple forms (as discussed by Leonhard Euler in the mid-1700s); other rational powers of  $e$  and tangents do not appear to. The sequence of odd numbers gives the continued fraction for  $\text{Coth}[1]$ ; the sequence of even numbers for  $\text{Bessel}[0, 1]/\text{Bessel}[1, 1]$ . In general, continued fractions whose  $n^{\text{th}}$  term is  $a n + b$  correspond to numbers given by  $\text{Bessel}[b/a, 2/a]/\text{Bessel}[b/a + 1, 2/a]$ . Numbers whose continued fraction terms are polynomials in  $n$  can presumably also be represented in terms of suitably generalized hypergeometric functions. (All so-called Hurwitz numbers have continued fractions that consist of interleaved polynomial sequences—a property left unchanged by  $x \rightarrow (ax + b)/(cx + d)$ .)

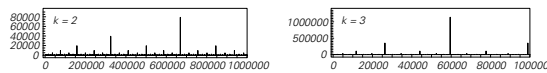
As discovered by Jeffrey Shallit in 1979, numbers of the form  $\text{Sum}[1/k^{2^i}, \{i, 0, \infty\}]$  that have nonzero digits in base  $k$  only at positions  $2^i$  turn out to have continued fractions with terms of limited size, and with a nested structure that can be found using a substitution system according to

```
{0, k - 1, k + 2, k, k, k - 2, k, k + 2, k - 2, k} //  
Nest[Flatten[{{1, 2}, {3, 4}, {5, 6}, {7, 8}, {5, 6}, {3, 4},  
{9, 10}, {7, 8}, {9, 10}, {3, 4}}] &], 1, n] //
```

The continued fractions for square roots are always periodic; for higher roots they never appear to show any significant regularities. The first million terms in the continued fraction for  $2^{1/3}$  contain 414,983 1's, have geometric mean 2.68505, and have largest term 4,156,269 at position 484,709. Terms of any size presumably in the end always occur in continued fractions for higher roots, though this is not known for certain. Fairly large terms are sometimes seen quite early: in  $5^{1/3}$  term 19 is 3052, while in  $\text{Root}[10 + 8\sqrt{-3} \&, 1]$  term 34

is 1,501,790. The presence of a large term indicates a close approximation to a rational number. In a few known cases simple formulas yield numbers that are close but not equal to integers. An example discovered by Srinivasa Ramanujan around 1913 is  $\text{Exp}[\pi\sqrt{163}]$ , which is an integer to one part in  $10^{30}$ , and has second continued fraction term 1,333,462,407,511. (This particular example can be understood from the fact that as  $d$  increases  $\text{Exp}[\pi\sqrt{d}]$  becomes extremely close to  $-1728 \text{KleinInvariantJ}[(1 + \sqrt{-d})/2]$ , which turns out to be an integer whenever there is unique factorization of numbers of the form  $a + b\sqrt{-d}$ —and  $d = 163$  is the largest of the 9 cases for which this is so.) Other less spectacular examples include  $\text{Exp}[\pi] - \pi$  and  $163/\text{Log}[163]$ .

Numbers with digits given by concatenation sequences in any base  $k$  (see note above) seem to have unusual continued fractions, in which most terms are fairly small, but some are extremely large. Thus with  $k = 2$ , term 30 is 4,534,532, term 64 is 4,682,854,730,443,938, term 152 is about  $2 \times 10^{34}$  and term 669,468 is about  $2 \times 10^{78902}$ . (For the  $k = 10$  case of the original Champernowne number, even term 18 is already about  $5 \times 10^{165}$ .) The plots below of the numbers of digits in successive terms turn out to have patterns of peaks that show some signs of nesting.



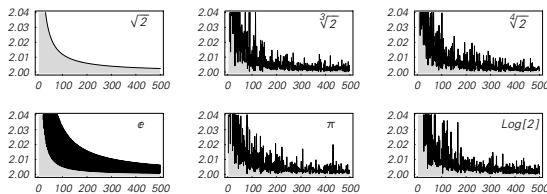
In analogy to digits in a concatenation sequence the terms in the sequence

$$\text{Flatten}[\text{Table}[\text{Rest}[\text{ContinuedFraction}[a/b], \{b, 2, n\}, \{a, b - 1\}]]]$$

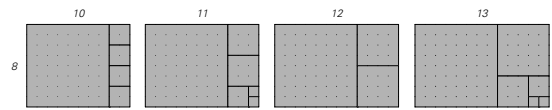
are known to occur with the same frequencies as they would in the continued fraction representation for a randomly chosen number.

The pictures below show as a function of  $n$  the quantity  $\text{With}[\{r = \text{FromContinuedFraction}[\text{ContinuedFraction}[x, n]]\}, -\text{Log}[\text{Denominator}[r], \text{Abs}[x - r]]]$

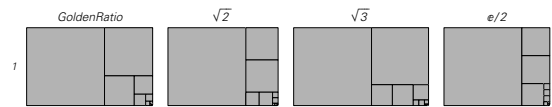
which gives a measure of the closeness of successive rational approximations to  $x$ . For any irrational number this quantity cannot be less than 2, while for algebraic irrationals Klaus Roth showed in 1955 that it can only have finitely many peaks that reach above any specified level.



■ **History.** Euclid's algorithm states that starting from integers  $\{a, b\}$  iterating  $\{a, b\} \rightarrow \text{If}[a > b, \{a - b, b\}, \{a, b - a\}]$  eventually leads to  $\{\text{GCD}[a, b], 0\}$ . (See page 1093.) The pictures below show how this works. The numbers of successively smaller squares (corresponding to the numbers of steps in the algorithm) turn out to be exactly  $\text{ContinuedFraction}[a/b]$ .



It was discovered in antiquity that Euclid's algorithm starting with  $\{x, 1\}$  terminates only when  $x$  is rational. In all cases, however, the relationship with continued fractions remains, as below.



Infinite continued fractions appear to have first been explicitly written down in the mid-1500s, and to have become popular in many problems in number theory by the 1700s. Leonhard Euler studied many continued fractions, while Joseph Lagrange seems to have thought that it might be possible to recognize any algebraic number from its continued fraction. The periodicity of continued fractions for quadratic irrationals was proved by Evariste Galois in 1828. From the late 1800s interest in continued fractions as such waned; it finally increased again in the 1980s in connection with problems in dynamical systems theory.

■ **Egyptian fractions.** Following the ancient Egyptian number system, rational numbers can be represented by sums of reciprocals, as in  $3/7 = 1/3 + 1/11 + 1/231$ . With suitable distinct integers  $a[n]$  one can represent any number by  $\text{Sum}[1/a[n], \{n, \infty\}]$ . The representation is not unique;  $a[n] = 2^n, n(n + 1)$  and  $(n + 1)!/n$  all yield 1. Simple choices for  $a[n]$  yield many standard transcendental numbers:  $n!$ ;  $e - 1$ ;  $n!^2$ ;  $\text{BesselJ}[0, 2] - 1$ ;  $n2^n$ ;  $\text{Log}[2]$ ;  $n^2$ ;  $\pi^2/6$ ;  $(2n - 1)(2n - 3)$ ;  $\pi\sqrt{3}/9$ ;  $3 - 16n + 16n^2$ ;  $\pi/8$ ;  $nn!$ ;  $\text{ExpIntegralEi}[1] - \text{EulerGamma}$ . (See also page 902.)

■ **Nested radicals.** Given a list of integers acting like digits one can consider representing numbers in the form  $\text{Fold}[\text{Sqrt}[\#1 + \#2] \&, 0, \text{Reverse}[\text{list}]]$ . A sequence of identical digits  $d$  then corresponds to the number  $(1 + \text{Sqrt}[4d + 1])/2$ . (Note that  $\text{Nest}[\text{Sqrt}[\# + 2], 0, n] = 2 \text{Cos}[\pi/2^{n+1}]$ .) Repeats of a digit block  $b$  give numbers that solve  $\text{Fold}[\#1^2 - \#2 \&, x, b] = x$ . It appears that digits 0, 1, 2 are



sufficient to represent uniquely all numbers between 1 and 2. For any number  $x$  the first  $n$  digits are given by

$$\text{Ceiling}[\text{NestList}[(2 - \text{Mod}[-\#, 1])^2 \&, x^2, n - 1] - 2]$$

Even rational numbers such as  $3/2$  do not yield simple digit sequences. For random  $x$ , digits 0, 1, 2 appear to occur with limiting frequencies  $\text{Sqrt}[2 + d] - \text{Sqrt}[1 + d]$ .

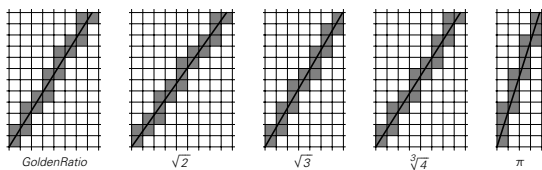
■ **Digital slope representation.** One can approximate a line of any slope  $h$  as in the picture below by a sequence of segments on a square grid (such as a digital display device). The vertical distance moved at the  $n^{\text{th}}$  horizontal position is  $\text{Floor}[nh] - \text{Floor}[(n-1)h]$ , and the sequence obtained from this (which contains only terms  $\text{Floor}[h]$  and  $\text{Floor}[h+1]$ ) provides a unique representation for  $h$ . As discussed on page 903 this sequence can be generated by applying substitution rules derived from the continued fraction form of  $h$ . If  $h$  is rational, the sequence is repetitive, while if  $h$  is a quadratic irrational, it is nested. Given a sequence of length  $n$ , an approximation to  $h$  can be reconstructed using

$$\text{Max}[\text{MapIndexed}[\#1/\text{First}[\#2] \&, \text{FoldList}[\text{Plus}, \text{First}[\text{list}], \text{Rest}[\text{list}]]]]$$

The fractional part of the result obtained is always an element of the Farey sequence

$$\text{Union}[\text{Flatten}[\text{Table}[a/b, \{b, n\}, \{a, 0, b\}]]]$$

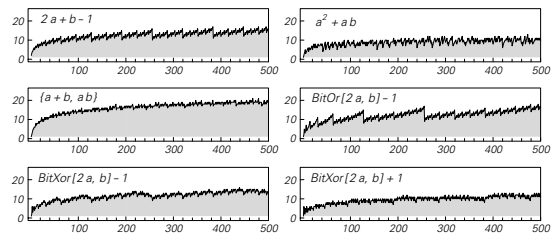
(See also pages 892, 932 and 1084.)



■ **Representations for integers.** See page 560.

■ **Operator representations.** Instead of repeatedly applying an operation to a sequence of digits one can consider forming integers (or other numbers) by performing trees of operations on a single constant. Thus, for example, any integer  $m$  can be obtained by a tree of  $m-1$  additions of 1's such as  $(1+(1+1))+1$ . Another operator that can be used to generate any integer is  $a \circ b = 2a + b - 1$ . In this case 6 is  $(1 \circ (1 \circ 1)) \circ 1$ , and an integer  $m$  can be obtained by  $\text{Tr}[1 + \text{IntegerDigits}[m, 2]] - 2$  or at most  $\text{Log}[2, m]$  applications of  $\circ$ . The operator  $ka + b - k + 1$  can be used for any  $k$ . It also turns out that  $\text{BitXor}[2a, b] + 1$  works, though in this case even for 2 the smallest representation is  $(1 \circ 1) \circ (1 \circ ((1 \circ 1) \circ 1))$ . (For  $\text{BitOr}[2a, b] - 1$  the number of applications needed is  $\text{With}[\{i = \text{IntegerDigits}[m, 2]\}, \text{Tr}[i + 1 + i[\#2]] (1 + i[\#3]) - 1]$ .) The pictures below show the smallest number of operator applications required for successive integers. With the pair of operators  $a + b$  and  $a \times b$  (a case considered in recreational

mathematics for  $n$ -ary operators) numbers of the form  $3^s$  have particularly small representations. Note that in all cases the size of the smallest representation must at some level increase like  $\text{Log}[m]$  (compare pages 1067 and 1070), but there may be some "algorithmically simple" integers that have shorter representations.



■ **Number classification.** One can imagine classifying real numbers in terms of what kinds of operations are needed to obtain them from integers. Rational numbers require only division (or solving linear equations), while algebraic numbers require solving polynomial equations. Rather little is known about numbers that require solving transcendental equations—and indeed it can even be undecidable (see page 1138) whether two equations can yield the same number. Starting with integers and then applying arithmetic operations and fractional powers one can readily reproduce all algebraic numbers up to degree 4, but not beyond. The sets of numbers that can be obtained by applying elementary functions like  $\text{Exp}$ ,  $\text{Log}$  and  $\text{Sin}$  seem in various ways to be disjoint from algebraic numbers. But if one applies multivariate elliptic or hypergeometric functions it was established in the late 1800s and early 1900s that one can in principle reach any algebraic number. One can also ask what numbers can be generated by integrals (or by solving differential equations). For rational functions  $f[x]$ ,  $\text{Integrate}[f[x], \{x, 0, 1\}]$  must always be a linear function of  $\text{Log}$  and  $\text{ArcTan}$  applied to algebraic numbers ( $f[x] = 1/(1+x^2)$  for example yields  $\pi/4$ ). Multiple integrals of rational functions can be more complicated, as in

$$\begin{aligned} &\text{Integrate}[1/(1+x^2+y^2), \{x, 0, 1\}, \{y, 0, 1\}] = \\ &\text{HypergeometricPFQ}[\{1/2, 1, 1\}, \{3/2, 3/2\}, 1/9]/6 + \\ &1/2 \pi \text{ArcSinh}[1] - \text{Catalan} \end{aligned}$$

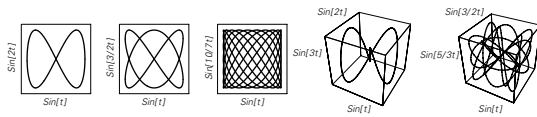
and presumably often cannot be expressed at all in terms of standard mathematical functions. Integrals of rational functions over regions defined by polynomial inequalities have recently been discussed under the name "periods". Many numbers associated with  $\text{Zeta}$  and  $\text{Gamma}$  can readily be generated, though apparently for example  $e$  and  $\text{EulerGamma}$  cannot. One can also consider numbers obtained from infinite sums (or by solving recurrence

equations). If  $f[n]$  is a rational function,  $Sum[f[n], \{n, \infty\}]$  must just be a linear combination of *PolyGamma* functions, but again the multivariate case can be much more complicated.

**Mathematical Functions**

■ **Page 145 · Mathematical functions.** (See page 1091.) *BesselJ*[0, x] goes like  $Sin[x]/\sqrt{x}$  for large x while *AiryAi*[-x] goes like  $Sin[x^{3/2}]/x^{1/4}$ . Other standard mathematical functions that oscillate at large x include *JacobiSN* and *MathieuC*. Most hypergeometric-type functions either increase or decrease exponentially for large arguments, though in the directions of Stokes lines in the complex plane they can oscillate sinusoidally. (For *AiryAi*[x] the Stokes lines are in directions  $(-1)^{\wedge}\{1, 2, 3\}/3$ .)

■ **Lissajous figures.** Plotting multiple sine functions each on different coordinate axes yields so-called Lissajous or Bowditch figures, as illustrated below. If the coefficients inside all the sine functions are rational, then going from  $t = 0$  to  $t = 2\pi$  Apply[LCM, Map[Denominator, list]] yields a closed curve. Irrational ratios of coefficients lead to curves that never close and eventually fill space uniformly.



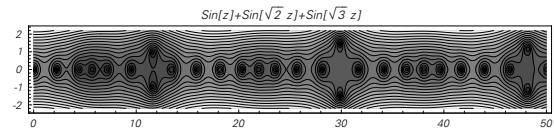
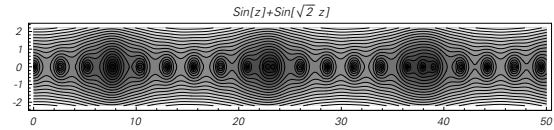
■ **Page 146 · Two sine functions.**  $Sin[ax] + Sin[bx]$  can be rewritten as  $2 Sin[1/2(a+b)x] Cos[1/2(a-b)x]$  (using *TrigFactor*), implying that the function has two families of equally spaced zeros:  $2\pi n/(a+b)$  and  $2\pi(n+1/2)/(b-a)$ .

■ **Differential equations.** The function  $Sin[x] + Sin[\sqrt{2}x]$  can be obtained as the solution of the differential equation  $y''[x] + 2y[x] - Sin[x] = 0$  with the initial conditions  $y[0] = 0, y'[0] = 2$ .

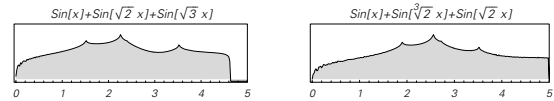
■ **Musical chords.** In a so-called equal temperament scale the 12 standard musical notes that make up an octave have a progression of frequencies  $2^{n/12}$ . Most schemes for musical tuning use rational approximations to these numbers. Until the past century, and since at least the 1300s, diminished fifth or tritone chords that consist of two notes (such as C and Gb) with frequency ratio  $\sqrt{2}$  have generally been avoided as sounding discordant. (See also page 1079.)

■ **Page 146 · Three sine functions.** All zeros of the function  $Sin[ax] + Sin[bx]$  lie on the real axis. But for  $Sin[ax] + Sin[bx] + Sin[cx]$ , there are usually zeros off the

real axis (even say for  $a = 1, b = 3/2, c = 5/3$ ), as shown in the pictures below.



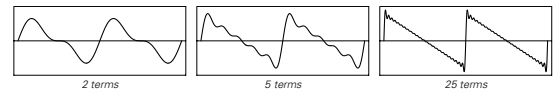
If a, b and c are rational,  $Sin[ax] + Sin[bx] + Sin[cx]$  is periodic with period  $2\pi/GCD[a, b, c]$ , and there are a limited number of different spacings between zeros. But in a case like  $Sin[x] + Sin[\sqrt{2}x] + Sin[\sqrt{3}x]$  there is a continuous distribution of spacings between zeros, as shown on a logarithmic scale below. (For  $0 < x < 10^6$  there are a total of 448,494 zeros, with maximum spacing  $\approx 4.6$  and minimum spacing  $\approx 0.013$ .)



■ **Page 147 · Substitution systems.**  $Cos[ax] - Cos[bx]$  has two families of zeros:  $2\pi n/(a+b)$  and  $2\pi n/(b-a)$ . Assuming  $b > a > 0$ , the number of zeros from the second family which appear between the  $n^{\text{th}}$  and  $(n+1)^{\text{th}}$  zero from the first family is  $(Floor[(n+1)\#] - Floor[n\#] \&)[(b-a)/(a+b)]$

and as discussed on page 903 this sequence can be obtained by applying a sequence of substitution rules. For  $Sin[ax] + Sin[bx]$  a more complicated sequence of substitution rules yields the analogous sequence in which  $-1/2$  is inserted in each *Floor*.

■ **Many sine functions.** Adding many sine functions yields a so-called Fourier series (see page 1074). The pictures below show  $Sum[Sin[nx]/n, \{n, k\}]$  for various numbers of terms k. Apart from a glitch that gets narrower with increasing k (the so-called Gibbs phenomenon), the result has a simple triangular form. Other so-called Fourier series in which the coefficient of  $Sin[mx]$  is a smooth function of m for all integer m yield similarly simple results.



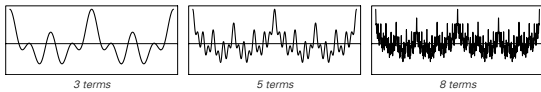
The pictures below show  $Sum[Sin[n^2 x]/n^2, \{n, k\}]$ , where in effect all coefficients of  $Sin[mx]$  other than those where m is

a perfect square are set to zero. The result is a much more complicated curve. Note that for  $x$  of the form  $p\pi/q$ , the  $k = \infty$  sum is just

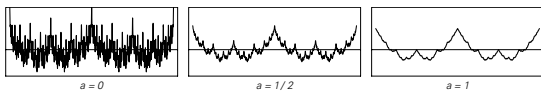
$$(\pi/(2q))^2 \text{Sum}[\text{Sin}[n^2 p \pi/q]/\text{Sin}[n \pi/(2q)]^2, \{n, q-1\}]$$



The pictures below show  $\text{Sum}[\text{Cos}[2^n x], \{n, k\}]$  (as studied by Karl Weierstrass in 1872). The curves obtained in this case show a definite nested structure, in which the value at a point  $x$  is essentially determined directly from the base 2 digit sequence of  $x$ . (See also page 1080.)



The curves below are approximations to  $\text{Sum}[\text{Cos}[2^n x]/2^{an}, \{n, \infty\}]$ . They can be thought of as having dimensions  $2-a$  and smoothed power spectra  $\omega^{-(1+2a)}$ .



■ **FM synthesis.** More complicated curves can be obtained for example using FM synthesis, as discussed on page 1079.

■ **Page 148 · Zeta function.** For real  $s$  the Riemann zeta function  $\text{Zeta}[s]$  is given by  $\text{Sum}[1/n^s, \{n, \infty\}]$  or  $\text{Product}[1/(1-\text{Prime}[n]^s), \{n, \infty\}]$ . The zeta function as analytically continued for complex  $s$  was studied by Bernhard Riemann in 1859, who showed that  $\text{PrimePi}[n]$  could be approximated (see page 909) up to order  $\sqrt{n}$  by  $\text{LogIntegral}[n] - \text{Sum}[\text{LogIntegral}[n^{r[i]}], \{i, -\infty, \infty\}]$ , where the  $r[i]$  are the complex zeros of  $\text{Zeta}[s]$ . The Riemann Hypothesis then states that all  $r[i]$  satisfy  $\text{Re}[r[i]] = 1/2$ , which implies a certain randomness in the distribution of prime numbers, and a bound of order  $\sqrt{n} \text{Log}[n]$  on  $\text{PrimePi}[n] - \text{LogIntegral}[n]$ . The Riemann Hypothesis is also equivalent to the statement that a bound of order  $\sqrt{n} \text{Log}[n]^2$  exists on  $\text{Abs}[\text{Log}[\text{Apply}[\text{LCM}, \text{Range}[n]]] - n]$ .

The picture in the main text shows  $\text{RiemannSiegelZ}[t]$ , defined as  $\text{Zeta}[1/2 + it] \text{Exp}[i \text{RiemannSiegelTheta}[t]]$ , where  $\text{RiemannSiegelTheta}[t] = \text{Arg}[\text{Gamma}[1/4 + it/2]] - 1/2 t \text{Log}[\pi]$

The first term in an approximation to  $\text{RiemannSiegelZ}[t]$  is  $2 \text{Cos}[\text{RiemannSiegelTheta}[t]]$ ; to get results to a given precision requires summing a number of terms that

increases like  $\sqrt{t}$ , making routine computation possible up to  $t \sim 10^{10}$ .

It is known that:

- The average spacing between zeros decreases like  $1/\text{Log}[t]$ .
- The amplitude of wiggles grows with  $t$ , but more slowly than  $t^{0.16}$ .
- At least the first 10 billion zeros have  $\text{Re}[s] = 1/2$ .

The statistical distribution of zeros was studied by Andrew Odlyzko and others starting in the late 1970s (following ideas of David Hilbert and George Pólya in the early 1900s), and it was found that to a good approximation, the spacings between zeros are distributed like the spacings between eigenvalues of random unitary matrices (see page 977).

In 1972 Sergei Voronin showed that  $\text{Zeta}[z + (3/4 + it)]$  has a certain universality in that there always in principle exists some  $t$  (presumably in practice usually astronomically large) for which it can reproduce to any specified precision over say the region  $\text{Abs}[z] < 1/4$  any analytic function without zeros.

### Iterated Maps and the Chaos Phenomenon

■ **History of iterated maps.** Newton's method from the late 1600s for finding roots of polynomials (already used in specific cases in antiquity) can be thought of as a smooth iterated map (see page 920) in which a rational function is repeatedly applied (see page 1101). Questions of convergence led in the late 1800s and early 1900s to interest in iteration theory, particularly for rational functions in the complex plane (see page 933). There were occasional comments about complicated behavior (notably by Arthur Cayley in 1879) but no real investigation seems to have been made. In the 1890s Henri Poincaré studied so-called return maps giving for example positions of objects on successive orbits. Starting in the 1930s iterated maps were sometimes considered as possible models in fields like population biology and business cycle theory—usually arising as discrete annualized versions of continuous equations like the Verhulst logistic differential equation from the mid-1800s. In most cases the most that was noted was simple oscillatory behavior, although for example in 1954 William Ricker iterated empirical reproduction curves for fish, and saw more complex behavior—though made little comment on it. In the 1950s Paul Stein and Stanislaw Ulam did an extensive computer study of various iterated maps of nonlinear functions. They concentrated on questions of convergence, but nevertheless noted complicated behavior. (Already in the

late 1940s John von Neumann had suggested using  $x \rightarrow 4x(1-x)$  as a random number generator, commenting on its extraction of initial condition digits, as mentioned on page 921.) Some detailed analytical studies of logistic maps of the form  $x \rightarrow ax(1-x)$  were done in the late 1950s and early 1960s—and in the mid-1970s iterated maps became popular, with much analysis and computer experimentation on them being done. But typically studies have concentrated on repetition, nesting and sensitive dependence on initial conditions—not on more general issues of complexity.

In connection with his study of continued fractions Carl Friedrich Gauss noted in 1799 complexity in the behavior of the iterated map  $x \rightarrow \text{FractionalPart}[1/x]$ . Beginning in the late 1800s there was number theoretical investigation of the sequence  $\text{FractionalPart}[a^n x]$  associated with the map  $x \rightarrow \text{FractionalPart}[ax]$  (see page 903), notably by G. H. Hardy and John Littlewood in 1914. Various features of randomness such as uniform distribution were established, and connections to smooth iterated maps emerged after the development of symbolic dynamics in the late 1930s.

■ **History of chaos theory.** See page 971.

■ **Page 150 • Exact iterates.** For any integer  $a$  the  $n^{\text{th}}$  iterate of  $x \rightarrow \text{FractionalPart}[ax]$  can be written as  $\text{FractionalPart}[a^n x]$ , or equivalently  $1/2 - \text{ArcTan}[\text{Cot}[a^n \pi x]]/\pi$ . In the specific case  $a = 2$  the iterates of  $\text{If}[x < 1/2, ax, a(1-x)]$  have the form  $\text{ArcCos}[\text{Cos}[2^n \pi x]]/\pi$ . (See pages 903 and 1098.)

■ **Page 151 • Problems with computer experiments.** The defining characteristic of a system that exhibits chaos is that on successive steps the system samples digits which lie further and further to the right in its initial condition. But in a practical computer, only a limited number of digits can ever be stored. In *Mathematica*, one can choose how many digits to store (and in the pictures shown in the main text, enough digits were used to avoid the problems discussed in this note). But a low-level language such as FORTRAN, C or Java always stores a fixed number of digits, typically around 53, in its standard double-precision floating-point representation of numbers.

So what happens when a system one is simulating tries to sample digits in its initial conditions beyond the ones that are stored? The answer depends on the way that arithmetic is handled in the computer system one uses.

When doing high-precision arithmetic, *Mathematica* follows the principle that it should only ever give digits that are known to be correct on the basis of the input that was provided. This means that in simulating chaotic systems, the numbers produced will typically have progressively fewer digits: later digits cannot be known to be correct without more precise knowledge of this initial condition.

(An example is `NestList[Mod[2#, 1] &, N[ $\pi/4$ , 40], 200]; Map[Precision, list]` gives the number of significant digits of each element in the list.)

But most current languages and hardware systems follow a rather different approach. (For low-precision machine arithmetic, *Mathematica* is also forced to follow this approach.) What they do is to give a fixed number of digits as the result of every computation, whether or not all those digits are known to be correct. It is then the task of numerical analysis to establish that in a particular computation, the final results obtained are not unduly affected by digits that are not known to be correct. And in practice, for many kinds of computations, this is to a large extent the case. But whenever chaos is involved, it is inevitably not.

As an example, consider the iterated map  $x \rightarrow \text{Mod}[2x, 1]$  discussed in the main text. At each step, this map shifts all the base 2 digits in  $x$  one position to the left. But if the computer gives a fixed number of digits at each step, then additional digits must be filled in on the right. On most computers, these additional digits are always 0. And so after some number of steps, all the digits in  $x$  are 0, and thus the value of  $x$  is simply 0.

But it turns out that a typical pocket calculator gives a different result. For pocket calculators effectively represent numbers in base 10 (actually so-called binary-coded decimal) not base 2, and fill in unknown digits with 0 in base 10. (Base 10 is used so that multiplying for example  $1/3$  by 3 gives exactly 1 rather than the more confusing result 0.9999... obtained with base 2.)

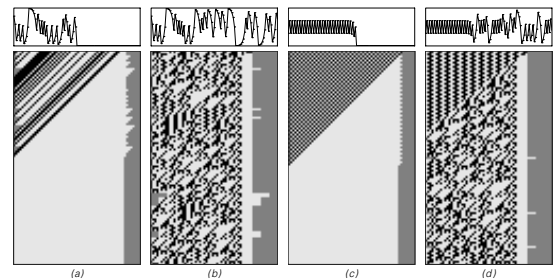
Pictures (a) and (c) below show simulations of the shift map on a typical computer, while pictures (b) and (d) show corresponding simulations on a pocket calculator. (Starting with initial condition  $x$  the digit sequence at step  $n$  is essentially

$$\text{IntegerDigits}[\text{Mod}[2^n \text{Floor}[2^{53} x], 2^{53}], 2, 53]$$

on the computer, and

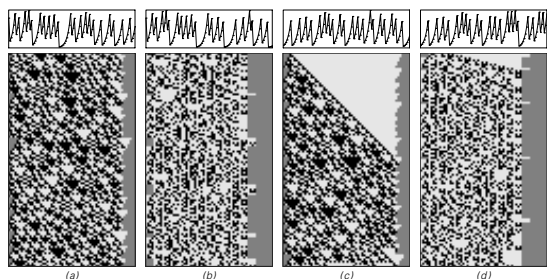
$$\text{Flatten}[\text{IntegerDigits}[\text{IntegerDigits}[\text{Mod}[2^n \text{Floor}[10^{12} x], 10^{12}], 10, 12], 2, 4]]$$

on the calculator. In both cases the limited number of digits implies behavior that ultimately repeats—but only long after the other effects we discuss have occurred.)



For the first several steps, the results as shown at the top of each corresponding picture agree. But as soon as the effect of sampling beyond the digits explicitly stored in the initial condition becomes important, the results are completely different. The computer gives simply 0, but the pocket calculator yields apparently random sequences—which turn out to be analogous to those discussed on page 319.

Other chaotic systems have a similar sensitivity to the details of computer arithmetic. But the simple behavior of the shift map turns out to be rather rare: in most cases—such as the multiplication by  $3/2$  shown in the pictures below—apparent randomness is produced, even on a typical computer.



It is important to realize however that this randomness has little to do with the details of the initial conditions. Instead, just as in other examples in this book, the randomness arises from an intrinsic process that occurs even with the simple repetitive initial condition shown in pictures (c) and (d) above.

Computer simulations of chaotic systems have been done since the 1950s. And it has often been observed that the sequences generated in these simulations look quite random. But as we now see, such randomness cannot in fact be a consequence of the chaos phenomenon and of sensitive dependence on initial conditions.

Nevertheless, confusingly enough, even though it does not come from sensitive dependence on initial conditions, such randomness is what makes the overall properties of simulations typically follow the idealized mathematical predictions of chaos theory. The point is that the presence of randomness makes the system behave on different steps as if it were evolving from slightly different initial conditions. But statistical averages over different initial conditions typically yield essentially the results one would get by evolution from a single initial condition containing an infinite number of randomly chosen digits.

■ **Page 152 · Mathematical perspectives.** Mathematicians may be confused by my discussion of complexity in iterated maps.

The first point to make is that the issues I am studying are rather different from the ones that are traditionally studied in the mathematics of these systems. The next point is that I have specifically chosen not to make the idealizations about numbers and operations on numbers that are usually made in mathematics.

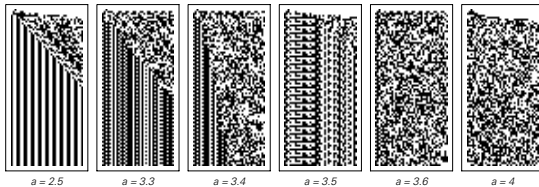
In particular, it is usually assumed that performing some standard mathematical operation, such as taking a square root, cannot have a significant effect on the system one is studying. But in trying to track down the origins of complex behavior, the effects of such operations can be significant. Indeed, as we saw on page 141, taking square roots can for example generate seemingly random digit sequences.

Many mathematicians may object that digit sequences are just too fragile an entity to be worth studying. They may argue that it is only robust and invariant concepts that are useful. But robustness with respect to mathematical operations is a different issue from robustness with respect to computational operations. Indeed, we will see later in this book that large classes of digit sequences can be considered equivalent with respect to computational operations, but these classes are quite different ones from those that are considered equivalent with respect to mathematical operations.

■ **Information content of initial conditions.** Common sense suggests that it is a quite different thing to specify a simple initial condition containing, say, a single black cell on a white background, than to specify an initial condition containing an infinite sequence of randomly chosen cells. But in traditional mathematics no distinction is usually made between these kinds of specifications. And as a result, mathematicians may find it difficult to understand my distinction between randomness generated intrinsically by the evolution of a system and randomness from initial conditions (see page 299). The distinction may seem more obvious if one considers, for example, sequential substitution systems or cyclic tag systems. For such systems cannot meaningfully be given infinite random initial conditions, yet they can still perfectly well generate highly random behavior. (Their initial conditions correspond in a sense to integers rather than real numbers.)

■ **Smooth iterated maps.** In the main text, all the functions used as mappings consist of linear pieces, usually joined together discontinuously. But the same basic phenomena seen with such mappings also occur when smooth functions are used. A particularly well-studied example (see page 918) is the so-called logistic map  $x \rightarrow ax(1-x)$ . The base 2 digit

sequences obtained with this map starting from  $x = 1/8$  are shown below for various values of  $a$ . The quadratic nature of the map typically causes the total number of digits to double at each step. But at least for small  $a$ , progressively more digits on the left show purely repetitive behavior. As  $a$  increases, the repetition period goes through a series of doublings. The detailed behavior is different for every value of  $a$ , but whenever the repetition period is  $2^j$ , it turns out that with any initial condition the leftmost digit always eventually follows a sequence that consists of repetitions of step  $j$  in the evolution of the substitution system  $\{1 \rightarrow \{1, 0\}, 0 \rightarrow \{1, 1\}\}$  starting either from  $\{0\}$  or  $\{1\}$ . As  $a$  approaches 3.569946, the period doublings get closer and closer together, and eventually a point is reached at which the sequence of leftmost digits is no longer repetitive but instead corresponds to the nested pattern formed after an infinite number of steps in the evolution of the substitution system. (An important result discovered by Mitchell Feigenbaum in 1975 is that this basic setup is universal to all smooth maps whose functions have a single hump.) When  $a$  is increased further, there is usually no longer repetitive or nested behavior. And although there are typically some constraints, the behavior obtained tends to depend on the details of the digit sequence of the initial conditions. In the special case  $a = 4$ , it turns out that replacing  $x$  by  $\text{Sin}[\pi u]^2$  makes the mapping become just  $u \rightarrow \text{FractionalPart}[2u]$ , revealing simple shift map dependence on the initial digit sequence. (See pages 1090 and 1098.)



■ **Higher-dimensional generalizations.** One can consider so-called Anosov maps such as  $\{x, y\} \rightarrow \text{Mod}[m . \{x, y\}, 1]$  where  $m$  is a matrix such as  $\{\{2, 1\}, \{1, 1\}\}$ . Any initial condition containing only rational numbers will then yield repetitive behavior, much as in the shift map. But as soon as  $m$  itself contains rational numbers, complicated behavior can be obtained even with an initial condition such as  $\{1, 1\}$ .

■ **Distribution of chaotic behavior.** For iterated maps, unlike for discrete systems such as cellular automata, one can get continuous ranges of rules by varying parameters. With maps based on piecewise linear functions the regions of parameters in which chaotic behavior occurs typically have simple shapes; with maps based, say, on quadratic

functions, however, elaborate nested shapes can occur. (See page 934.)

■ **Page 155 · Lyapunov exponents.** The number of new digits that are affected at each step by a small change in initial conditions gives the so-called Lyapunov exponent  $\lambda$  for the evolution. After  $t$  steps, the difference in size resulting from the change in initial conditions will be multiplied by approximately  $2^{\lambda t}$ —at least until this difference is of order 1. (See page 950.)

■ **Chaos in nature.** See page 304.

■ **Bitwise operations.** Cellular automata can be thought of as analogs of iterated maps in which bitwise operations such as *BitXor* are used instead of ordinary arithmetic ones. (See page 906.)

**Continuous Cellular Automata**

■ **Implementation.** The state of a continuous cellular automaton at a particular step can be represented by a list of numbers, each lying between 0 and 1. This list can then be updated using

```
CCAEvolveStep[f_, list_List] :=
  Map[f, (RotateLeft[list] + list + RotateRight[list])/3]
CCAEvolveList[f_, init_List, t_Integer] :=
  NestList[CCAEvolveStep[f, #] &, init, t]
```

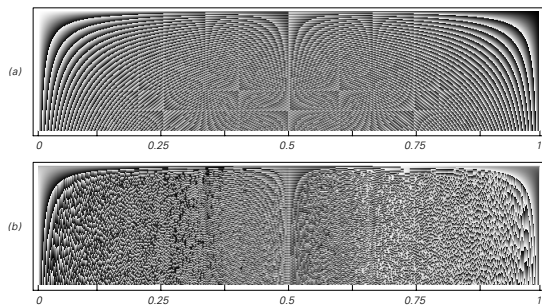
where for the rule on page 157  $f$  is  $\text{FractionalPart}[3\#/2]$  & while for the rule on page 158 it is  $\text{FractionalPart}[\# + 1/4]$  &.

Note that in the definitions above, the elements of *list* can be either exact rational numbers, or approximate numbers obtained using  $N$ . For rough calculations, standard machine-precision numbers may sometimes suffice, but for detailed calculations exact rational numbers are essential. Indeed, the presence of exponentially increasing errors would make the bottom of the picture on page 157 qualitatively wrong if just 64-bit double-precision numbers had been used. On page 160 the effect is much larger, and almost all the pictures would be completely wrong—with the notable exception of the one that shows localized structures.

■ **History.** Continuous cellular automata have been introduced independently several times, under several different names. In all cases the rules have been at least slightly more complicated than the ones I consider here, and behavior starting from simple initial conditions does not appear to have been studied before. Versions of continuous cellular automata arose in the mid-1970s as idealizations of coupled ordinary differential equations for arrays of nonlinear oscillators, and implicitly in finite difference approximations to partial differential equations. They began

to be studied with extensive computer simulations in the early 1980s, probably following my work on ordinary cellular automata. Most often considered, notably by Kunihiko Kaneko and co-workers, were so-called “coupled map lattices” or “lattice dynamical systems” in which an iterated map (typically a logistic map) was applied at each step to a combination of neighboring cell value. A transition from regular class 2 to irregular class 3 behavior, with class 4 behavior involving localized structures in between, was observed, and was studied in detail by Hugues Chaté and Paul Manneville, starting in the late 1980s.

■ **Page 158 · Properties.** At step  $t$  the background is  $FractionalPart[at]$ . For rational  $a$  this always repeats, cycling through  $Denominator[a]$  possible values (compare page 255). In most patterns generated from initial conditions containing say a single black cell most cells whose values are not forced to be the same end up being at least slightly different—even in cases like  $a = 0.375$ . Note that in cases like  $a = 0.475$  there is some trace of a pattern at every step—but it only becomes obvious when it makes values wrap around from 1 to 0. The pictures below show successive colors of (a) the background (compare page 950) and (b) the center cell for each  $a = n/500$  from 0 to 1 for the systems on page 159. (Compare page 243.)

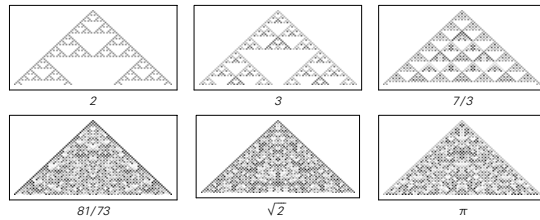


If  $a$  is not a rational number the background never repeats, but the main features of patterns obtained seem similar.

■ **Additive rules.** In the case  $a = 0$  the systems on page 159 are purely additive. A simpler example is the rule

$$\text{Mod}[\text{RotateLeft}[\text{list}] + \text{RotateRight}[\text{list}], 1]$$

With a single nonzero initial cell with value  $1/k$  the pattern produced is just Pascal’s triangle modulo  $k$ . If  $k$  is a rational number only a limited set of values appear, and the pattern has a nested form analogous to those shown on page 870. If  $k$  is irrational then equidistribution of  $\text{Mod}[\text{Binomial}[t, x], k]$  implies that all possible values eventually appear; the corresponding patterns seem fairly irregular, as shown below. (Compare pages 953 and 1092.)



■ **Probabilistic cellular automata.** As an alternative to having continuous values at each cell, one can consider ordinary cellular automata with discrete values, but introduce probabilities for, say, two different rules to be applied at each cell. Examples of probabilistic cellular automata are shown on page 591; their behavior is typically quite similar to continuous cellular automata.

### Partial Differential Equations

■ **Ordinary differential equations.** It is also possible to set up systems which have a finite number of continuous variables (say  $a[t]$ ,  $b[t]$ , etc.) that change continuously with time. The rules for such systems correspond to ordinary differential equations. Over the past century, the field of dynamical systems theory has produced many results about such systems. If all equations are of the form  $a'[t] = f[a[t], b[t], \dots]$ , etc. then it is known for example that it is necessary to have at least three equations in order to get behavior that is not ultimately fixed or repetitive. (The Lorenz equations are an example.) If the function  $f$  depends explicitly on time, then two equations suffice. (The van der Pol equations are an example.)

Just as in iterated maps, a small change in the initial values  $a[0]$  etc. can often lead to an exponentially increasing difference in later values of  $a[t]$ , etc. But as in iterated maps, the main part of this process that has been analyzed is simply the excavation of progressively less significant digits in the number  $a[0]$ .

(Note that numerical simulations of ODEs on computers must approximate continuous time by discrete steps, making the system essentially an iterated map, and often yielding spurious complicated behavior.)

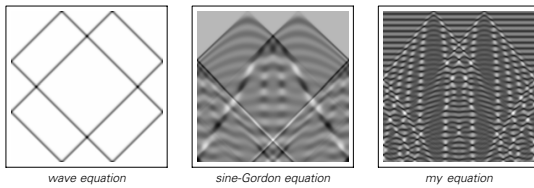
■ **Klein-Gordon equation.** The behavior of the Klein-Gordon equation  $\partial_{tt}u[t, x] = \partial_{xx}u[t, x] - u[t, x]$  is visually very similar to that shown for the sine-Gordon equation. For the Klein-Gordon equation, however, there is an exact solution:

$$u[t, x] = \text{If}[x^2 > t^2, 0, \text{BesselJ}[0, \text{Sqrt}[t^2 - x^2]]]$$

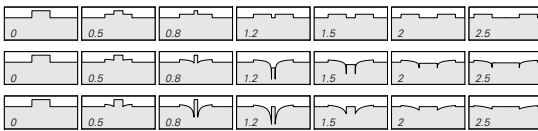
■ **Origins of the equations.** The diffusion equation arises in physics from the evolution of temperature or of gas density.

The wave equation represents the propagation of linear waves, for example along a compressible spring. The sine-Gordon equation represents nonlinear waves obtained for example as the limit of a very large number of pendulums all connected to a spring. The traditional name of the equation is a pun on the Klein-Gordon equation that appears in relativistic quantum mechanics and in describing strings in elastic media. It is notable that unlike with ODEs, essentially all PDEs that have been widely studied come quite directly from physics. My PDE on page 165 is however an exception.

■ **Nonlinearity.** The pictures below show behavior with initial conditions containing two Gaussians (and periodic boundary conditions). The diffusion and wave equations are linear, so that results are linear sums of those with single Gaussians. The sine-Gordon equation is nonlinear, but its solutions satisfy a generalized linear superposition principle. The equation from page 165 shows no such simple superposition principle. Note that even with a linear equation, fairly complicated patterns of behavior can sometimes emerge as a result of boundary conditions.



■ **Higher dimensions.** The pictures below show as examples the solution to the wave equation in 1D, 2D and 3D starting from a stationary square pulse.



In each case a 1D slice through the solution is shown, and the solution is multiplied by  $r^{d-1}$ . For the wave equation, and for a fair number of other equations, even and odd dimensions behave differently. In 1D and 3D, the value at the origin quickly becomes exactly 0; in 2D it is given by  $1-t/\text{Sqrt}[t^2-1]$ , which tends to zero only like  $-1/(2t^2)$  (which means that a sound pulse cannot propagate in a normal way in 2D).

■ **Page 164 · Singular behavior.** An example of an equation that yields inconsistent behavior is the diffusion equation with a negative diffusion constant:

$$\partial_t u[t, x] = -\partial_{xx} u[t, x]$$

This equation makes any variation in  $u$  as a function of  $x$  eventually become infinitely rapid.

Many equations used in physics can lead to singularities: the Navier-Stokes equations for fluid flow yield shock waves, while the Einstein equations yield black holes. At a physical level, such singularities usually indicate that processes not captured by the equations have become important. But at a mathematical level one can simply ask whether a particular equation always has solutions which are at least as regular as its initial conditions. Despite much work, however, only a few results along these lines are known.

■ **Existence and uniqueness.** Unlike systems such as cellular automata, PDEs do not have a built-in notion of “evolution” or “time”. Instead, as discussed on page 940, a PDE is essentially just a constraint on the values of a function at different times or different positions. In solving a PDE, one is usually interested in determining values that satisfy this constraint inside a particular region, based on information about values on the edges. It is then a fundamental question how much can be specified on the edges in order to obtain a unique solution. If too little is specified, there may be many possible solutions, while if too much is specified there may be no consistent solution at all. For some very simple PDEs, the conditions for unique solutions are known. So-called hyperbolic equations (such as the wave equation, the sine-Gordon equation and my equation) work a little like cellular automata in that in at least one dimension information can propagate only at a limited speed, say  $c$ . The result is that in such equations, giving values for  $u[t, x]$  at  $t = 0$  for  $-s < x < s$  will uniquely determine  $u[t, x]$  at larger  $t$  for  $-s + ct < x < s - ct$ . In other PDEs, such as so-called elliptic ones, there is no such limit on the rate of information propagation, and as a result, it is immediately necessary to know values of  $u[t, x]$  at all  $x$ , and on the boundaries of the region, in order to determine  $u[t, x]$  for any  $t > 0$ .

■ **Page 165 · Field equations.** Any equation of the form

$$\partial_{tt} u[t, x] = \partial_{xx} u[t, x] + f[u[t, x]]$$

can be thought of as a classical field equation for a scalar field. Defining

$$v[u] = -\text{Integrate}[f[u], u]$$

the field then has Lagrangian density

$$((\partial_t u)^2 - (\partial_x u)^2)/2 - v[u]$$

and conserves the Hamiltonian (energy function)

$$\text{Integrate}[(\partial_t u)^2 + (\partial_x u)^2]/2 + v[u], [x, -\infty, \infty]$$

With the choice for  $f[u]$  made here (with  $a \geq 0$ ),  $v[u]$  is bounded from below, and as a result it follows that no singularities ever occur in  $u[t, x]$ .



■ **Equation for the background.** If  $u[t, x]$  is independent of  $x$ , as it is sufficiently far away from the main pattern, then the partial differential equation on page 165 reduces to the ordinary differential equation

$$u''[t] = (1 - u[t]^2)(1 + a u[t])$$

$$u[0] = u'[0] = 0$$

For  $a = 0$ , the solution to this equation can be written in terms of Jacobi elliptic functions as

$$\sqrt{3} \operatorname{JacobiSN}[t/3^{1/4}, 1/2]^2 / (1 + \operatorname{JacobiCN}[t/3^{1/4}, 1/2]^2)$$

In general the solution is

$$b d \operatorname{JacobiSN}[r t, s]^2 / (b - d \operatorname{JacobiCN}[r t, s]^2)$$

where

$$r = -\operatorname{Sqrt}[1/8 a c (b - d)]$$

$$s = d (c - b) / (c (d - b))$$

and  $b, c, d$  are determined by the equation

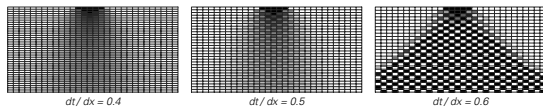
$$(x - b)(x - c)(x - d) = -(12 + 6 a x - 4 x^2 - 3 a x^3) / (3 a)$$

In all cases (except when  $-8/3 < a < -1/\sqrt{6}$ ), the solution is periodic and non-singular. For  $a = 0$ , the period is  $2 \cdot 3^{1/4} \operatorname{EllipticK}[1/2] \approx 4.88$ . For  $a = 1$ , the period is about 4.01; for  $a = 2$ , it is about 3.62; while for  $a = 4$ , it is about 3.18. For  $a = 8/3$ , the solution can be written without Jacobi elliptic functions, and is given by

$$3 \operatorname{Sin}[\operatorname{Sqrt}[5/6] t]^2 / (2 + 3 \operatorname{Cos}[\operatorname{Sqrt}[5/6] t]^2)$$

■ **Numerical analysis.** To find numerical solutions to PDEs on a digital computer one has no choice but to make approximations. In the typical case of the finite difference method one sets up a system with discrete cells in space and time that is much like a continuous cellular automaton, and then hopes that when the cells in this system are made small enough its behavior will be close to that of the continuous PDE.

Several things can go wrong, however. The pictures below show as one example what happens with the diffusion equation when the cells have size  $dt$  in time and  $dx$  in space. So long as the so-called Courant condition  $dt/dx < 1/2$  is satisfied, the results are correct. But when  $dt/dx$  is made larger, an instability develops, and the discrete approximation yields completely different results from the continuous PDE.



Many methods beyond finite differences have been invented over the past 30 years for finding numerical solutions to PDEs. All however ultimately involve discretization, and can suffer from difficulties that are similar—though often more insidious—to those for finite differences.

For equations where one can come at least close to having explicit algebraic formulas for solutions, it has often been possible to prove that a certain discretization procedure will yield correct results. But when the form of the true solution is more complicated, such proofs are typically impossible.

And indeed in practice it is often difficult to tell whether complexity that is seen is actually a consequence of the underlying PDE, or is instead merely a reflection of the discretization procedure. I strongly suspect that many equations, particularly in fluid dynamics, that have been studied over the past few decades exhibit highly complex behavior. But in most publications such behavior is never shown, presumably because the authors are not sure whether the behavior is a genuine consequence of the equations they are studying.

■ **Implementation.** All the numerical solutions shown were found using the *NDSolve* function built into *Mathematica*. In general, finite difference methods, the method of lines and pseudospectral methods can be used. For equations of the form

$$\partial_{tt} u[t, x] = \partial_{xx} u[t, x] + f[u[t, x]]$$

one can set up a simple finite difference method by taking  $f$  in the form of pure function and creating from it a kernel with space step  $dx$  and time step  $dt$ :

$$\text{PDEKernel}[f_, \{dx_, dt_\}] := \text{Compile}[\{a, b, c, d\}, \text{Evaluate}[(2 b - d) + ((a + c - 2 b)/dx^2 + f[b]) dt^2]]$$

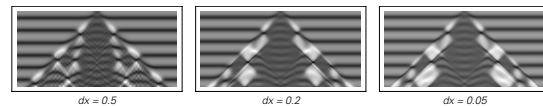
Iteration for  $n$  steps is then performed by

$$\begin{aligned} \text{PDEEvolveList}[ker_, \{u0_, u1_, n_\}] := \\ \text{Map}[\text{First}, \text{NestList}[\text{PDEStep}[ker, \#] \& \{u0, u1\}, n]] \\ \text{PDEStep}[ker_, \{u1_, u2_\}] := \{u2, \text{Apply}[ker, \text{Transpose}[\{\text{RotateLeft}[u2], u2, \text{RotateRight}[u2], u1\}], \{1\}]\} \end{aligned}$$

With this approach an approximation to the top example on page 165 can be obtained from

$$\begin{aligned} \text{PDEEvolveList}[\text{PDEKernel}[ \\ (1 - \#^2)(1 + \#) \&, \{0.1, 0.05\}], \text{Transpose}[ \\ \text{Table}[\{1, 1\} \text{N}[\text{Exp}[-x^2]], \{x, -20, 20, 0.1\}], 400] \end{aligned}$$

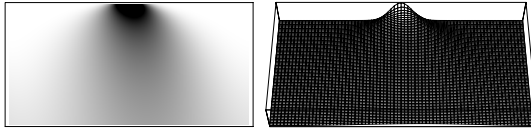
For both this example and the middle one the results converge rapidly as  $dx$  decreases. But for the bottom example, the pictures below show that convergence is not so rapid, and indeed, as is typical in working with PDEs, despite having used large amounts of computer time I do not know whether the details of the picture in the main text are really correct. The energy function (see above) is at least roughly conserved, but it seems quite likely that the “shocks” visible are merely a consequence of the discretization procedure used.



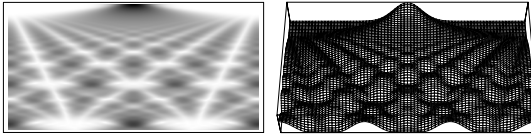
■ **Different powers.** The equations

$$\partial_{tt} u[t, x] = \partial_{xx} u[t, x] + (1 - u[t, x]^n)(1 + a u[t, x])$$

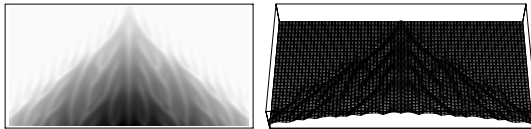
with  $n = 4, 6, 8$ , etc. appear to show similar behavior to the  $n = 2$  equation in the main text.



Burger's equation:  $\partial_t u[t, x] = \partial_{xx} u[t, x] - u[t, x] \partial_x u[t, x]$



nonlinear Schrödinger equation:  $i \partial_t u[t, x] = -\partial_{xx} u[t, x] + 4 \text{Abs}[u[t, x]]^2 u[t, x]$



Kuramoto-Sivashinsky equation:  $\partial_t u[t, x] = -\partial_{xx} u[t, x] - 1/2 \partial_{xxxx} u[t, x] + (\partial_x u[t, x])^2$

■ **Other PDEs.** The pictures above show three PDEs that have been studied in recent years. All are of the so-called parabolic type, so that, unlike my equation, they have no

limit on the rate of information propagation, and thus a solution in any region immediately depends on values on the boundary—which in the pictures below is taken to be periodic. (The deterministic Kardar-Parisi-Zhang equation  $\partial_t u[t, x] = a \partial_{xx} u[t, x] + 1/2 b (\partial_x u[t, x])^2$  yields behavior like Burger's equation, but symmetrical. Note that  $\text{Abs}[u]$  is plotted in the second picture, while for the last equation a common less symmetrical form replaces the last term by  $u[t, x] \partial_x u[t, x]$ .)

**Continuous Versus Discrete Systems**

■ **History.** From the late 1600s when calculus was invented it took about two centuries before mathematicians came to terms with the concepts of continuity that it required. And to do so it was necessary to abandon concrete intuition, and instead to rely on abstract mathematical theorems. (See page 1149.) The kind of discrete systems that I consider in this book allow a return to a more concrete form of mathematics, without the necessity for such abstraction.

■ **"Calculus".** It is an irony of language that the word "calculus" now associated with continuous systems comes from the Latin word which means a small pebble of the kind used for doing discrete calculations (same root as "calcium").